NOVA / AX PRODUCT DISCOVERY STUDIO

Project Context Document for Coding Environments

Version 3.0 | February 2026 | Confidential

0. How to Use This Document

This document is the master context file for the Nova/AX codebase. It is designed to be fed into AI coding assistants (Claude Code, Codex, Cursor, etc.) so that any agent working on this project understands the full picture: what we are building, who the users are, what terminology means, what belongs to this week vs the future, and what architectural boundaries exist between Nova as a platform and client projects like 2gthr.

When starting a new coding session, reference this document alongside the relevant module specification. This document answers 'what is the world this code lives in' while module specs answer 'what exactly should this module do'.

1. Glossary of Key Terms

Naming confusion is the biggest risk in this project. The following definitions are canonical and must be used consistently in all code, comments, variable names, database schemas, and documentation.

1.1 Platform & Business Entity Terms

Term	Definition
AX	The company Daniel owns. AX is the legal entity that owns the Nova platform IP. AX sells services and platform access to clients like Accenture. In the codebase: AX is the parent brand. Sometimes referred to as 'AX Product Discovery Studio'.
Nova	The software platform that AX is building. Nova is a multi-tenant, AI-powered platform that accelerates the entire product development lifecycle: discovery, validation, development, launch, measurement, and continuous optimisation. Nova doesn't just help teams know what to build; it actively helps them build it, measure it, and improve it through agentic automation. In the codebase: 'Nova' is the application name and the name users see in the UI.
2gthr (Together)	A product being built by Accenture for their client (a South African mutual/insurance company). 2gthr is a membership-driven, AI-enabled life partner for ambitious South African professionals. 2gthr is NOT a Nova module. It is the end-product that Nova is helping to discover, validate, and build. In the codebase: 2gthr data, configs, and assets must be scoped under the client project, never baked into Nova's core.
Forward Deployed Engineer (FDE)	A Nova-certified consultant (often embedded in the client's team) who uses Nova to run product development processes. The FDE

	configures Nova for the client, uploads client data, runs validation sessions, manages the client's workspace, and leverages Nova's agentic capabilities. In the codebase: FDE is a user role with elevated permissions within a client workspace.
Nova Plugin	An injectable code package that Nova generates for a client's frontend application. The plugin collects anonymised usage analytics and serves contextual feedback prompts to end-users. It is the bridge between the client's live product and Nova's intelligence layer. In the codebase: this is a standalone, embeddable JS/React package with its own build pipeline.
Feature ID	A unique identifier assigned to every mapped feature/functionality in a client's system. Feature IDs allow Nova to track analytics, feedback, and product decisions at the feature level. Everything in Nova's intelligence loop ties back to Feature IDs. In the codebase: featureId is a string field present on AnalyticsEvent, FeedbackResponse, BacklogItem, and Feature entities.

1.2 User Hierarchy Terms

Nova has a complex user hierarchy because agencies (like Accenture) use Nova to serve their own clients. This creates a three-tier structure.

Term	Definition	Example
Nova Client (Tenant)	The company paying for Nova access. A Tenant in the multi-tenancy model. Has a billing relationship with AX.	Accenture South Africa
Client Project	A specific engagement/product the Nova Client is working on. A Tenant can have multiple Projects. Not all Tenant employees have access to all Projects.	2gthr
End Client	The Nova Client's customer who owns the product being built. May have limited, permissioned access to the Nova Project workspace.	The mutual/insurance company commissioning 2gthr
End User / Target User	The real person the End Client's product is being built for. They never log into Nova directly. However, they may interact with Nova indirectly through the Nova Plugin (feedback prompts, community, public backlog).	An ambitious South African professional using 2gthr

Feedback User	A person giving structured feedback on solutions. They access Nova through shared links or through the Nova Plugin embedded in the client's product. They build profiles over time.	A participant in a validation survey or in-app feedback prompt
Community User	An early adopter invited to co-design the product. Has access to the Early Adopters Community, Public Backlog, and deeper product strategy discussions. Recruited through the Plugin or direct invitation.	A beta tester who opted in via the plugin's community CTA

1.3 Nova Platform User Roles (within a Project)

Role	Description	Access Level
Platform Admin	AX team member. Full system access across all tenants.	God mode
Tenant Admin	Senior person at the Nova Client. Manages users, projects, billing.	Full within tenant
Project Owner	Leads a project. Manages team, modules, parameters.	Full within project
FDE	Forward Deployed Engineer. Configures Nova, manages data. May span projects.	Full operational. Clear project indicator.
Product Team Member	Works within a project (e.g., Ray). Uses modules, reviews analytics and feedback.	Module-level per Project Owner
Client Viewer	End Client stakeholder. View access.	View-only on permitted modules
Client Contributor	End Client stakeholder. Edit access to specific areas.	Edit on permitted, view elsewhere
Leadership Voter	Stakeholder invited to vote via shared links.	Voting sessions + Nova overview
Feedback User	External person. Feedback via links or Plugin.	Specific sessions only
Community User	Early adopter. Community space and public backlog.	Community, backlog, limited visibility

1.4 2gthr-Specific Terms (Client Project Context)

These terms belong to the 2gthr product domain, NOT to Nova's architecture. They should only appear in 2gthr project data.

Term	Definition (2gthr Context)
MI (Mutual Intelligence)	The AI companion at the heart of 2gthr. Deeply personalised conversational co-pilot.
MyDNA	Persistent identity layer. Three profiles: Functional, Growth, Behavioural.
Paths	Medium-to-long-term journeys. Types: To-Do, Aspiration, Cluster, Life Shock.
Milestones	Meaningful checkpoints within Paths. Typically 3-18 months.
Go-Cards	Atomic units of action. Types: Event, Tool, Narrative, Media, Connect, Reflection, Evidence, etc.
Strides	Progress Strides (personal growth) and Impact Strides (contribution).
Circles	Peer communities. Profession-based or interest-based.
LifeMap	Member's home view of all Paths across five life dimensions.
Five Life Dimensions	Career, Financial, Wellness, Life Experience, Impact.

2. What Nova Is

2.1 The Full Vision

Nova is an AI-powered platform that accelerates the entire product development lifecycle. It does not just help teams figure out what to build. It helps them build it, launch it, measure it, and continuously improve it through agentic automation.

The platform creates a compounding intelligence loop: the more context a client feeds Nova (architecture, design systems, user data, market intelligence, live analytics, user feedback), the more Nova can do. At maturity, Nova becomes the operating system for product development.

2.2 Nova's Value Chain

Stage 1: Know What to Build (Discovery & Validation)

- Research Engine: Deep research on topics, markets, competitors, user needs.
- Insight Discovery: Transform research into structured, actionable insights.
- Problem Discovery: Convert insights into scored problem/opportunity statements.
- Problem Prioritisation: Interactive voting and evidence-based leadership decisions.
- Solution Design: Conceptual solutions with assumption mapping and feasibility.
- Validation Engine: Design and run experiments before committing resources.

Stage 2: Build It Right (Development Acceleration)

- Frontend Code Generation: POCs and production components in the client's design language.
- Business Requirements Generation: BRDs, user stories, acceptance criteria.
- Data Schema Design: Data objects derived from frontend components.
- Code Testing & QA: Automated testing, evals, guardrails.

Stage 3: Launch It Smart (Analytics & Measurement)

- Analytics Injection: Embed analytics into every frontend feature for day-one measurement.
- Nova Plugin Deployment: Injectable code collecting anonymised usage data and serving feedback prompts.
- Usage Metrics & Dashboards: Real product usage tied to Feature IDs flowing back into Nova.
- Market Intelligence: Ongoing competitive monitoring tied to product decisions.

Stage 4: Improve It Continuously (The Intelligence Loop)

- System Audit & Feature Mapping: Ingest the client's system, map every feature with unique Feature IDs.
- Contextual Feedback: Plugin dynamically builds feedback sessions based on the user's feature/journey context.
- Early Adopters Community: Deeper product strategy discussions with engaged end-users.

- Public Backlog: Voteable, commentable feature roadmap for community users.
- Insight-to-Action Loops: Usage + feedback + market trends flow back into Nova.
- AI-Suggested Improvements: Nova's solution bot incorporates all evidence streams and suggests changes.

Stage 5: Automate the Work (AI Strategy & Agents)

- PDLC Automation Mapping: Identify which lifecycle parts can be augmented/automated.
- Agent Orchestration: Deploy and manage AI agents across the PDLC.
- Design Generation: Nova produces mockups in the client's design language for review.
- Automated Development: With permissions, Nova branches code, implements, documents, and pushes for approval.

Stage 6: Scale the Intelligence (Platform Maturity)

- Cross-Project Learning: Patterns and best practices improve across all projects.
- Continuous Intelligence: Understanding deepens with every interaction.
- Exportable Outputs: All outputs are client-owned and exportable.

2.3 Nova's Module Architecture

Module	Purpose	Status
Client Onboarding	Ingest everything about the client's product.	V1 manual (FDE)
System Audit & Feature Map	Ingest client's system. Map every feature with unique Feature IDs. Foundation for all intelligence modules.	BUILD NOW
Target Audience	Audience segments, personas, TAM/SAM/SOM.	Needs MVP
Market Intelligence	Competitive intel, market research, usage benchmarks.	Minimal
Research Engine	Deep research. Multi-source AI synthesis.	V1 (agents)
Insight Discovery	Structured insights. Dashboards.	V1 (agents)
Problem Discovery	Problem statements. Scoring, dedup, evidence.	V1 (agents)
Problem Prioritisation	Interactive voting. Evidence trails.	Built
Solution Design	Conceptual solutions. Assumptions, feasibility.	V1 (agents)
Validation Engine	Experiment design. Cost/effort estimation.	V1 prompt
Dev Acceleration	POC building, frontend code gen, schemas, BRDs, testing.	Active

Analytics Layer	Embed analytics, track usage, decision dashboards.	Planned
Nova Plugin	Injectable frontend code. Anonymised analytics + contextual feedback. Client-controlled display rules.	BUILD NOW
Public Backlog	Voteable feature roadmap. Upvote/downvote with reasoning.	BUILD NOW
Community Hub	Early adopters community. Co-design participation.	Planned
Product Intelligence Console	Admin command centre. Usage + feedback + market trends + Nova suggestions. Interactive improvement sessions.	Planned
AI Strategy	Map automation opportunities. Deploy agents. AI strategy.	Vision

3. Product Intelligence & Continuous Improvement System

This is the largest and most differentiating capability Nova is building. It closes the loop between 'build a product' and 'continuously improve it based on real evidence'.

3.1 The Intelligence Loop

Five interconnected components form a continuous cycle:

1. System Audit & Feature Mapping: Nova ingests the client's system and creates a structured map with unique Feature IDs.
2. Nova Plugin: Code embedded in the client's frontend collects anonymised analytics and serves contextual feedback, all tied to Feature IDs.
3. Feedback & Community Layer: Contextual feedback, early adopters community, and public backlog.
4. Product Intelligence Console: Admin interface aggregating usage, feedback, and market trends, with Nova suggesting improvements.
5. Design & Development Automation: Nova generates designs, mockups, code changes, and pushes branches for review.

Each component feeds the next. The loop never stops.

3.2 Component 1: System Audit & Feature Mapping

What It Does

Nova ingests a client's existing system and produces a structured map of every feature. This is the foundation everything else builds on.

Inputs Nova Consumes

- Codebase (frontend and backend): Components, routes, features, user flows.
- BRDs and product documentation: Requirements, specs, user stories.
- Design files (Figma, Sketch, etc.): UI/UX mapped to features.
- Existing analytics: If available, ingested and mapped.
- Client commentary: Product team adds context, priorities, known issues.

What Nova Produces

- Feature Map: Every feature gets a unique Feature ID with purpose, location, related features, user flows, and current state.
- System Architecture View: Structured data showing how features relate, which journeys they serve, how data flows.
- Enriched Feature Profiles: Market insights and product intelligence layered onto each feature.

Feature ID Design Principles

- Feature IDs are referenced across all intelligence modules: analytics, feedback, backlog, console.
- Feature IDs link back to code locations, design files, and BRDs for full traceability.
- Feature IDs should be human-readable where possible (e.g., 'path-milestone-completion' not 'feat_0847').
- Hierarchical: features can have parent features and belong to journeys.

V1 Implementation

For the 2gthr POC, Feature IDs are created manually/semi-automated. Full automated code scanning is future scope.

3.3 Component 2: The Nova Plugin

What It Is

A standalone, injectable code package embedded in the client's frontend. It is Nova's 'eyes and ears' inside the live product.

Function A: Anonymised Usage Analytics

- Tracks user flows: which features used, order, duration, drop-offs.
- All data anonymised. No PII. Anonymous session IDs only.
- Events tagged with Feature IDs mapping to Nova's Feature Map.
- Data flows to Nova's Analytics Layer powering dashboards: adoption, churn, underuse.
- Lightweight and non-blocking. No degradation of client frontend performance.

Function B: Contextual Feedback Prompts

- Pop-up feedback prompts served at contextually appropriate moments.
- Dynamically generated based on which Feature IDs the user just interacted with.
- Feedback types: satisfaction ratings, NPS, open-text, feature-specific, comparative preferences.
- Users can always opt out. Opt-out respected and tracked.
- Opt-out flow includes CTA to join the Early Adopters Community.

Client-Controlled Display Rules

- Environment targeting: beta/staging only, production only, or both.
- Frequency caps: per session, per day, per week.
- Feature targeting: which Feature IDs trigger prompts.
- User segment targeting: all users, new users, power users, specific journey completers.
- Prompt timing: on completion, after X seconds, on exit intent, on specific actions.
- Visual customisation: style-able to match the client's design system.

Technical Architecture

- Standalone React/JS package with own build pipeline.
- Communicates with Nova backend via Firebase Functions API.

- Receives configuration (Feature IDs, prompts, rules) from Nova on init.
- V1 is React-specific (2gthr POC is React). Future: framework-agnostic.

V1 Scope (Build This Week)

- Basic anonymous event tracking tied to Feature IDs.
- Simple contextual feedback prompt (1-2 questions) after key journey completions.
- Opt-out flow with community CTA.
- Data pipeline: plugin events to Firebase, displayed in basic Nova dashboard.

3.4 Component 3: Feedback & Community Layer

Early Adopters Community

- Space for deeper product strategy discussions with engaged end-users.
- Users arrive via plugin community CTA or direct invitation.
- Curated insights about product direction. Structured feedback sessions.
- Early access to features and news.
- Profiles build over time, reducing repeated data collection.

Public Backlog

- Curated, public-facing feature roadmap for community users.
- Items linked to Feature IDs and/or Problem IDs from discovery modules.
- Shows: name, description, expected timing, status (planned/in-progress/shipped).
- Upvote/downvote with mandatory comment on downvotes.
- Votes and comments flow to Product Intelligence Console as structured data.
- Product team controls which items are public vs internal-only.

3.5 Component 4: Product Intelligence Console (Planned)

Admin command centre aggregating all intelligence streams:

Data Streams

- Usage Analytics: From plugin. Features used, adoption curves, drop-offs.
- User Feedback: From prompts, community, backlog votes/comments.
- Market Intelligence: From research and competitive modules.
- Problem & Solution History: Full evidence chain from discovery.

Nova's Active Intelligence

- Solution bot incorporates all streams and suggests feature-level changes.
- Admin does an interactive session: review data, see suggestions, make decisions.
- For approved changes, Nova generates designs in client's design language.
- On design approval, Nova generates implementation specs.
- With permissions: Nova branches repo, implements, documents, pushes for review.

The Admin Journey

6. Open Console for a feature or feature group.
7. See: usage metrics, feedback sentiment, market context, Nova suggestions.
8. Interactive session with Nova to refine changes.
9. Nova produces design mockups.
10. Admin reviews and tweaks.
11. On approval: Nova generates BRDs, schemas, acceptance criteria.
12. With dev permissions: Nova branches, implements, documents, pushes for code review.
13. Dev lead reviews. Changes merge. Plugin tracks updated feature performance.
14. Loop continues.

3.6 Current State & What Has Been Done

- Client system architecture and design system documented via offline bots.
- POC of new feature built in client's design language (React/Tailwind).
- POC launching this week: first opportunity to deploy the Nova Plugin.
- Feature IDs need to be defined for POC before plugin deployment.

4. This Week's Priorities

4.1 Primary Goal: Get Ray Using Nova

Ray should create a Nova profile and use the platform to do what he's been doing manually. First real user test.

4.2 Voting Module (Built)

Largely built. Remaining: footer on the voting page introducing Nova with a CTA to explore the platform. Do not rely on logo click. Footer must be intentional and visible.

4.3 Nova Plugin V1 (Build Now)

- Define Feature IDs for the POC's key features and flows.
- Build anonymous event tracking tied to Feature IDs.
- Build simple contextual feedback prompt after key journey completions.
- Build opt-out flow with community CTA.
- Build data pipeline: events to Firebase, basic Nova dashboard.

4.4 System Audit & Feature Map V1 (Build Now)

- Define Feature ID schema and naming convention.
- Map POC features, flows, and screens to Feature IDs.
- Store Feature Map in Firestore as project-level data.
- Make Feature IDs available to the plugin.

4.5 Public Backlog V1 (Build Now)

- Display items with name, description, status, expected timing.
- Upvote/downvote with mandatory comment on downvotes.
- Votes and comments flow to Nova.

4.6 V1 Module Shells (Should-Build)

Shells for all modules so Ray and leadership see the full platform scope.

4.7 Nova Home Page

All modules as tiles. Communicate the full value chain. Inaccessible modules show as 'coming soon'.

5. Multi-Tenancy Architecture

5.1 Three-Tier Model

Tier 1: Tenant

The paying organisation (e.g., Accenture). User directory, billing, project list.

Tier 2: Project

A specific engagement (e.g., 2gthr). Own data silo, team, roles, module configs. Primary isolation boundary.

Tier 3: Session/Feedback

Time-bounded interactions with scoped access for external users.

5.2 Data Isolation Rules

- Project data never leaks to other projects or Nova core.
- FDEs see unmissable project indicator. Wrong-project uploads are catastrophic.
- Client terminology is project-level config, not Nova vocabulary.
- Per-module, per-project, and per-session permissions.
- Plugin data (analytics, feedback) is project-scoped. API enforces isolation.

5.3 Generalisation Principle

- Voting module votes on generic 'problems'.
- Feature Map maps generic 'features'. For 2gthr: Paths, Go-Cards, MI.
- Plugin tracks generic 'feature interactions'.
- Public Backlog displays generic 'planned improvements'.
- Everything 2gthr-specific is project data, not platform architecture.

6. Key User Flows

6.1 Flow A: Ray's First Login

15. Ray creates account or is invited.
16. Assigned to 2gthr project as Product Team Member.
17. Lands on Nova home showing all modules.
18. Sees which modules have data vs shells.
19. Navigates to Problem Prioritisation, sees 50+ problems.

6.2 Flow B: Ray Creates a Voting Session

20. Selects problems for voting.
21. Configures session: title, deadline, method.
22. Generates shareable link.
23. Sends to leadership.
24. Monitors progress in real-time.

6.3 Flow C: Leadership Votes and Discovers Nova

25. Clicks voting link, lands on clean interface.
26. Votes on problems with evidence/context.
27. Footer introduces Nova with CTA to explore.
28. Clicks through to Nova overview.
29. Module tiles show full scope. Inaccessible = 'coming soon'.

6.4 Flow D: End User Interacts with Nova Plugin

30. 2gthr user completes a key journey in the POC.
31. Plugin detects completion via Feature ID tracking.
32. Plugin serves contextual prompt: 'How was your experience with [feature]?'
33. User responds or opts out.
34. Opt-out shows community CTA.
35. Response tagged with Feature IDs, sent to Nova.
36. Ray sees feedback + analytics in Nova dashboard.

6.5 Flow E: Community User and Public Backlog

37. Community user navigates to Public Backlog.
38. Sees upcoming features with descriptions and timelines.
39. Upvotes or downvotes (pushed reasoning on downvotes).
40. Leaves comments with suggestions.
41. Votes and comments flow into Nova intelligence.

6.6 Flow F: Admin Improvement Session (Future)

42. Admin opens Product Intelligence Console for a feature cluster.
43. Sees usage metrics, feedback, market context, Nova suggestions.
44. Interactive session with Nova to refine changes.
45. Nova produces design mockups.
46. Admin approves. Nova generates specs.
47. With permissions: Nova branches, implements, documents, pushes.
48. Dev lead reviews. Changes merge. Plugin tracks new performance. Loop continues.

7. Architecture Boundaries

7.1 Nova Core

- Auth, roles, tenant/project management.
- Module framework (routing, navigation, permissions, layout).
- Generic data models: Problem, Insight, Feature, VotingSession, FeedbackSession, BacklogItem, AnalyticsEvent, etc.
- UI component library.
- Agent orchestration layer.
- Analytics event ingestion pipeline.
- Plugin configuration and deployment system.
- Public Backlog rendering engine.

7.2 Project-Level Data

- All client content: problems, research, solutions, personas.
- Client domain models (2gthr: Paths, Go-Cards, MyDNA) as project data.
- Feature Map with Feature IDs.
- Plugin config: display rules, prompt templates, targeting.
- Analytics events and feedback responses.
- Public Backlog items and community data.
- Generated assets: code, designs, schemas, BRDs.

7.3 Not Built Yet

- Automated code scanning for feature mapping. V1 is manual.
- Full Product Intelligence Console. Planned.
- Automated git branching and code implementation. Vision.
- Feedback marketplace. Future.
- Psychometric tests. Future.
- Billing management. Manual.
- Full Community Hub with moderation. Basic for now.

8. Technical Stack & Constraints

Frontend: React with Tailwind CSS

Backend: Firebase (Auth, Firestore, Functions, Storage)

Nova Plugin: Standalone React/JS package. Own build pipeline. API via Firebase Functions.

Source Control: Git / Bitbucket

AI Agents: Claude, GPT, Gemini (via Firebase Functions)

Design Reference: Figma (MPC integration)

Coding Assistants: Claude Code, Codex (consume this document)

Constraints:

- No framework deviations from chosen stack.
- Reusable shared component library.
- Firebase security rules enforce multi-tenancy.
- Generic schemas; client data as project-level documents.
- Plugin embeddable in React frontends with minimal effort. V1 React-specific.
- Plugin analytics lightweight and non-blocking.
- All plugin data project-scoped. API rejects cross-project access.

9. 2gthr Platform Context (Agent Reference)

Client project content, NOT Nova architecture.

9.1 What 2gthr Is

Membership-driven, AI-enabled life partner for ambitious South African professionals. Five life dimensions: Career, Financial, Wellness, Life Experience, Impact. Targets doctors, lawyers, accountants, engineers. Addresses SA-specific challenges.

9.2 Core Architecture

Aspiration > Path > Milestone > Cluster > Go-Card. MI is the AI companion. MyDNA is the identity layer. Multiple simultaneous journeys, personalised by context.

9.3 Work Done

50+ problem statements. Leadership voted. 20 shortlisted. Solutions designed. Full POC built in React/Tailwind. POC launching this week with Nova Plugin.

10. Strategic Context (Decision-Making, Not Code)

10.1 Accenture Relationship

First client. Dual purpose: deliver 2gthr value AND prove Nova's commercial viability. Leadership determines budget.

10.2 Demo Opportunity

Every Nova interaction should communicate a vision far bigger than the immediate task. Voting footer, module shells, plugin capabilities all reinforce breadth.

10.3 Plugin as Sales Tool

The Nova Plugin is the most visceral demonstration of value. When leadership sees live usage analytics and user feedback flowing into Nova in real-time, the proposition moves from theoretical to undeniable.

10.4 IP Protection

Nova methodology, agents, frameworks, platform code = AX IP. Client deliverables (research, problems, solutions, generated code) = client IP. Plugin is AX IP; data collected = project client IP.

11. Appendix: Data Model Quick Reference

11.1 Platform-Level Entities (Nova Core)

Entity	Description	Key Fields
Tenant	Paying organisation.	id, name, plan, billing, settings
Project	Engagement within tenant.	id, tenantId, name, description, status, config
User	Nova account holder.	id, email, name, role, tenantIds, activeProjectId
ProjectMembership	User-project link.	userId, projectId, role, modulePermissions
Module	Functional area.	id, name, description, icon, status, order, valueChainStage

11.2 Discovery & Validation Entities (Project-Scoped)

Entity	Description	Key Fields
ResearchTopic	Topic being researched.	id, projectId, title, status, outputs
Insight	Structured finding.	id, projectId, topicId, statement, evidence, tags
Problem	Validated problem statement.	id, projectId, statement, score, evidence, status
SolutionConcept	Conceptual solution.	id, projectId, problemIds, description, assumptions, feasibility
VotingSession	Stakeholder voting round.	id, projectId, title, problemIds, voters, deadline, status, results
Vote	Individual vote.	id, sessionId, voterId, problemId, value, timestamp
Persona	Target user segment.	id, projectId, name, demographics, needs, behaviours

ValidationExperiment	Validation run.	id, projectId, solutionId, type, hypothesis, status, results

11.3 Product Intelligence Entities (Project-Scoped)

Entity	Description	Key Fields
Feature	Mapped feature in client's system.	id, projectId, name, description, type, parentFeatureId, journeyIds, codeLocation, designRef, brdRef, status
FeatureJourney	User flow connecting features.	id, projectId, name, featureIds (ordered), description
AnalyticsEvent	Usage event from plugin.	id, projectId, featureId, eventType, sessionId, timestamp, metadata
PluginConfig	Plugin display rules.	id, projectId, environment, frequencyCaps, targetFeatureIds, segmentRules, promptTemplates, styling
FeedbackPrompt	Configured feedback template.	id, projectId, triggerFeatureIds, triggerCondition, questions, active
FeedbackResponse	User's feedback.	id, projectId, promptId, featureIds, sessionId, responses, timestamp, optedOut
BacklogItem	Public backlog item.	id, projectId, featureId, problemId, title, description, status, expectedRelease, upvotes, downvotes
BacklogVote	Vote on backlog item.	id, backlogItemId, userId, direction, comment, timestamp
CommunityUser	Early adopter profile.	id, projectId, userId, joinSource, joinDate, participationHistory

GeneratedAsset	Nova-produced output.	id, projectId, type, moduleSource, featureId, content, exportable
AgentRun	AI agent execution record.	id, projectId, agentType, input, output, status, timestamp

--- End of Context Document ---

Feed this document into your coding environment at the start of each session.