# Comprehensive Strategic Blueprint: Award-Winning Kiroween Hackathon Concepts

## Executive Summary: The Agentic Strategy for Kiroween

This report presents four fully conceptualized, award-winning hackathon submissions, each rigorously tailored to one of the Kiroween categories and designed for optimal scoring under the Official Rules. The core methodology across all projects is the advanced utilization of the Kiro developer toolchain, emphasizing deep mastery of Model Context Protocol (MCP), Spec-Driven Development, and Agent Hooks to achieve a maximum score in the critical Implementation criterion.1 The concepts move beyond simple code generation, positioning Kiro as an intelligent orchestrator capable of automating high-complexity, domain-specific tasks.

The strategic framing prioritizes not only winning a Category Prize ($5,000) but also targeting the high-value Best Startup Project Bonus Prize ($10,000) for the most architecturally robust submissions, maximizing total potential reward.1

Table: Strategic Concept Alignment with Judging Criteria

| **Concept** | **Category** | **Maximized Criteria** | **Strategic Rationale** |
| --- | --- | --- | --- |
| Phoenix Proxy | Resurrection | Implementation (Kiro Use), Potential Value | Addresses high-cost technical debt; achieves peak Kiro leverage via a custom MCP server for legacy context. |
| Chronos Nexus | Frankenstein | Quality & Design, Potential Value | Unique resource integration (purged public data + niche commercial APIs); high societal/research value. |
| Nexus Blueprint | Skeleton Crew | Implementation, Potential Value | Solves the complex "two distinct apps" mandate with a versatile, modern architectural foundation (Event Sourcing). |
| Shadow Log | Costume Contest | Quality & Design (UI/UX) | Achieves "haunting" UI through functional design principles (Digital Brutalism) while enhancing security vigilance. |

## Section I: Kiroween Hackathon Strategic Compliance and Kiro Platform Mastery

### A. Judging Criteria Deconstruction and Scoring Strategy

The evaluation framework mandates success across three equally weighted criteria: Potential Value, Implementation, and Quality and Design.1 Achieving an award-winning entry requires a targeted strategy for each.

The **Potential Value** criterion assesses the broad usefulness, ease of use, accessibility, and scalability of the solution.1 The strategy involves focusing on real-world, high-cost industry pain points, such as legacy modernization (Phoenix Proxy) or the creation of novel economic and social insights from underutilized data (Chronos Nexus). The proposed solutions are designed for extensibility and clear market differentiation.1

The **Implementation** score is the most direct measure of expertise, requiring demonstrated, next-level understanding of Kiro features, variety of usage, and strategic decision-making in the development workflow.1 The four concepts presented herein mandate the use of all five core Kiro capabilities: Vibe Coding, Hooks, Steering Docs, Specs, and especially, custom Model Context Protocol (MCP) integration. Incorporating a custom MCP server is explicitly designed to show how Kiro's capabilities were extended to solve problems that would have been "difficult or impossible" otherwise, securing the highest possible score in this category.1

The **Quality and Design** criterion requires creativity, originality, and a polished final product, often involving the use of unique public resources or thoughtful UI/UX choices.1 This is addressed through the fusion of rare public datasets in the Frankenstein project 1 and the functionally enhanced, psychologically resonant user experience in the Costume Contest entry.1

### B. Essential Compliance Checklist and Technical Requirements

Strict adherence to the Official Rules is paramount, as failure to meet structural or timing requirements results in disqualification.1 The Submission Period closes on December 5, 2025, and all artifacts must be finalized by this deadline.1

The following compliance points are foundational:

1. **Repository Integrity:** The source code must be hosted in a public, open-source repository that includes an approved OSI Open Source License, which must be clearly visible.1
2. **Kiro Context Mandate:** Every project must contain the /.kiro directory at the project root to house the required steering, specs, and hooks files. Critically, this directory cannot be included in the .gitignore file.1
3. **Demonstration Video:** The accompanying demonstration video must be publicly accessible (e.g., YouTube or Vimeo) and must not exceed three (3) minutes in length. Judges are not required to watch beyond this duration.1
4. **Skeleton Crew Specific Rule:** The strategic requirement for the Skeleton Crew entry—to demonstrate two distinct applications from one foundation—must be materialized physically as two separate repository folders within the submission.1

Table: Hackathon Compliance and Submission Checklist

| **Requirement** | **Status** | **Notes/File Location** | **Source Rule** |
| --- | --- | --- | --- |
| Eligible Entrant/Team Max 3 | Confirmed | Representative assigned, eligibility verified. | Rule 3 |
| Submission Period Met | Confirmed | Deadline: Dec 5, 2025 (2:00 pm PT). | Rule 1 |
| Working Software Application | Mandatory | Link to website/demo provided for testing. | Rule 4 (Functionality, Testing) |
| Public Open Source Repository | Mandatory | Must include approved OSI Open Source License visible at the top. | Rule 4 (Submission Requirements) |
| Required Kiro Directory (/.kiro) | Mandatory | Must contain specs, hooks, and steering and NOT be ignored. | Rule 4 (Submission Requirements) |
| Demo Video < 3 Minutes | Mandatory | Uploaded to public platform (YouTube/Vimeo) without copyrighted music. | Rule 4 (Submission Requirements) |
| Two Separate Repos | Conditional (Skeleton Crew Only) | Required for Skeleton Crew submission only. | Rule 4 (Skeleton Crew projects only) |

A crucial element of the strategic planning is the understanding of prize stacking. An eligible project can win one Category Prize AND one Bonus Prize.1 By framing the structurally advanced projects (Resurrection or Skeleton Crew) as viable initial Software as a Service (SaaS) offerings, the project can potentially qualify for the **Best Startup Project Bonus Prize** ($10,000), which requires confirmation of founder status, a LinkedIn profile, and a startup website.1 This allows for a total prize pool of $15,000 (Category + Bonus), significantly exceeding the $5,000 base Category Prize, provided the team is eligible to submit as a startup.1

### C. Kiro Toolchain Advanced Utilization Blueprint

To achieve a top Implementation score, the integration of Kiro must move beyond simple Q&A assistance to demonstrate strategic orchestration of the development lifecycle.1

1. **Steering Documents (.kiro/steering/):** These documents provide Kiro with persistent architectural knowledge, ensuring consistency without repetitive prompts.5 Advanced use involves forcing Kiro to adhere to niche project constraints. For the Resurrection project, the tech.md steering file must specify arcane legacy system protocols and data structures. For the Costume Contest, steering enforces strict design token constraints, acting as a digital system guardian.6
2. **Spec-Driven Development (.kiro/specs/):** Specs formalize complex feature development through structured artifacts: requirements.md (user stories), design.md (architecture blueprint), and tasks.md (implementation checklist).4 The advanced application of Specs involves defining microservice domain boundaries (Skeleton Crew) or documenting complex, field-by-field data mappings for integration (Resurrection), demonstrating Kiro's role in planning and accountability, not just coding.7
3. **Agent Hooks:** Hooks are intelligent automation rules that trigger specific AI actions based on IDE events like file creation or saving.8 Moving beyond basic style formatting, advanced deployment includes Hooks for proactive system tasks, such as automated test synchronization, compliance checks, and schema validation. These Hooks ensure system consistency across complex, multi-component architectures and significantly reduce manual overhead.9
4. **Model Context Protocol (MCP) Integration:** MCP is the mechanism for achieving the highest level of implementation mastery by connecting Kiro to specialized external servers, data, or tools.10 The construction of a **custom MCP server** is the most potent strategic element. This custom server acts as a translator or specialized knowledge base, giving the AI access to domain-specific context (e.g., proprietary documentation or database schemas) it would otherwise lack. This capability directly satisfies the Implementation requirement to show how MCP enabled features or workflow improvements that would have been exceptionally difficult or impossible using Kiro’s default capabilities.1

## Section II: Concept 1: Resurrection - The Legacy COBOL API Gateway (Phoenix Proxy)

### A. The Challenge: Reviving Obsolete Business Logic

The Resurrection category demands reimagining obsolete technology with modern innovations.1 Legacy systems, particularly those relying on decades-old languages like COBOL, remain critical infrastructure across finance and government, yet they present severe challenges due to high maintenance costs, poor security integration, and complex data formats.13 These systems contain indispensable business logic but are functionally "dead" to modern development practices.15

### B. Proposed Solution: The "Phoenix Proxy"

The Phoenix Proxy is a spec-driven API gateway designed to automatically generate modern, secure API wrappers around existing legacy system interfaces (e.g., COBOL applications or mainframe transactions). This approach avoids costly and risky legacy code refactoring by creating a scalable, cloud-native proxy layer.13 The value proposition is profound: reducing technical debt and enabling fast, secure integration of mission-critical logic into modern systems.14

### C. Kiro Implementation Rationale (Maximizing Implementation Score via Custom MCP)

The complexity of legacy integration lies in the translation layer—mapping archaic, fixed-format data fields (like COBOL PIC clauses) to flexible, modern JSON/REST schemas. This mapping is highly context-sensitive.

1. **Custom MCP Integration (The COBOL Data Dictionary Mapper):** This feature is the strategic differentiator. A custom MCP server is developed to access and parse COBOL Data Dictionary files. When Kiro is generating the API wrapper code, it uses this MCP server tool to query the precise definition of legacy fields.10 The result is that Kiro can produce validated, secure, and accurate code that respects the byte-level offsets and constraints of the COBOL system. This level of granular, domain-specific context eliminates the common problem of "Workslop"—AI-generated code that appears functional but fails due to missing critical context—a frequent failure mode in highly specialized AI applications.17
2. **Spec-Driven Bridge Development:** The translation logic is formalized through Kiro Specs.7 The design.md specifies the required field mappings and data transformations (e.g., handling null values or date formats). Kiro generates the actual translation code for the modern microservice (the Proxy) based on the formal spec, validated against the context provided by the custom MCP server.4
3. **Agent Hooks for Regression Testing:** Legacy systems are inherently fragile. An Agent Hook is configured to trigger upon changes to the API mapping logic.8 The hook automatically updates or generates corresponding unit tests that assert the generated JSON output matches expected values derived from the legacy data structure. This automated test synchronization maintains functional integrity and high code quality in a volatile integration environment.9
4. **Steering Docs for Protocol Enforcement:** The tech.md steering document is used to enforce external requirements, such as specific security standards (e.g., mandatory header configurations for mainframe authentication) and stringent error handling procedures, compelling Kiro to integrate these non-standard rules consistently throughout the generated proxy code.6

## Section III: Concept 2: Frankenstein - The Archival Data Sentinel (Chronos Nexus)

### A. The Challenge: Reconciling Purged History with Current Reality

The Frankenstein category requires the integration of "seemingly incompatible elements to build something unexpectedly powerful".1 This concept stitches together two highly disparate data streams: sensitive, often politically purged, historical U.S. government open datasets (e.g., climate, environmental, and demographic data that disappeared from Data.gov after major administrative changes) 2 and proprietary, contemporary niche demographic and market research data (e.g., data from Niche or Statista).19

The power derived from this combination is the ability to restore lost context and perform long-term accountability analysis, providing novel insights into socioeconomic trends or infrastructure changes that traditional, time-restricted datasets cannot offer.21

### B. Proposed Solution: "The Chronos Nexus"

The Chronos Nexus is an AI-powered analytical platform that fuses these datasets into a unified, queryable data lake. The system’s primary value is its ability to answer complex, high-stakes research questions that bridge historical events (tracked in the purged data) with current market reality (tracked in the commercial data). For example, a user could analyze the long-term impact of deleted 2017 environmental enforcement data on current housing market trends in specific regions, a function currently hampered by information asymmetry.22

### C. Kiro Implementation Rationale (Focus on Vibe Coding and Data Integrity Hooks)

The project maximizes the Quality and Design criterion by leveraging "unique public resources like datasets" 1 and uses Kiro to manage the high variance inherent in archival data ingestion.

1. **Vibe Coding for Iterative Data Discovery:** Data archived from government repositories often has unstable or undocumented schemas, requiring extensive cleaning.2 The primary interface is Kiro’s Vibe Coding session.23 A user conversationally defines the required output (e.g., a time-series graph correlating two datasets), and Kiro rapidly generates the necessary Python scripts for data parsing, transformation, and visualization, allowing for fluid, conversational refinement of the research query.
2. **Agent Hooks for Data Integrity and Compliance:** Data integrity is a critical risk. An Agent Hook is configured to execute when any new historical or commercial dataset is introduced into the workspace.8 The hook runs a pre-defined schema validation process against an expected data model. If the schema is inconsistent (e.g., a required column is missing or improperly formatted), the hook alerts the developer and forces Kiro to generate a remedial data cleaning script before the file is accepted for ingestion.9
3. **Steering Docs for Ethical Governance:** The combination of sensitive public data and proprietary commercial data necessitates stringent control. The product.md steering document specifies strict data governance policies, licensing requirements, and mandatory citation formats. This forces Kiro to embed source attribution and usage constraints into any generated output or report, ensuring ethical data handling and compliance.6 This strategic use of steering supports the philosophical need for accountability by prioritizing accuracy over convenience, addressing inherent market failures in information control.22

## Section IV: Concept 3: Skeleton Crew - The Event-Sourced Microservice Foundation (Nexus Blueprint)

### A. The Challenge: Building a Lean, Versatile, Dual-Application Foundation

The Skeleton Crew category requires building a foundational template capable of supporting "two distinct applications" derived from the same skeleton, with the strict technical mandate that the repository must contain **two separate repo folders** for these applications.1

### B. Proposed Solution: "The Nexus Blueprint" (Kiro-Generated Event Sourcing Starter Kit)

The chosen architecture is the **Event Sourcing (ES) pattern** combined with Command Query Responsibility Segregation (CQRS).25 This architecture intrinsically meets the requirement by mandating separate services (microservices) that share a single, scalable event log (the "Skeleton"). The ES pattern is a highly versatile, foundational structure known for enabling loose coupling, resilient communication, and a complete audit trail.26

1. **The Foundation (Skeleton):** A core Event Store, which is the single source of truth, persisting all state changes as domain events (e.g., UserRegistered, ItemAdded).26
2. **Distinct Application 1 (Transactional Command Service):** A microservice responsible for transactional operations (e.g., an Inventory Management service). It receives user commands and publishes new events to the Event Store. This is stored in **Repository Folder A**.
3. **Distinct Application 2 (Analytical Query Dashboard):** A microservice responsible for read operations (e.g., a Real-Time Sales Dashboard). It subscribes to the same events and maintains a highly optimized, read-only data projection. This is stored in **Repository Folder B**.1

This architectural choice provides a professional-grade solution to the hackathon rule, demonstrating the template's immediate capability to support autonomous applications, thereby maximizing the Potential Value and extensibility criteria.1

### C. Kiro Implementation Rationale (Focus on Foundation Generation and Consistency)

Kiro is indispensable for automating the boilerplate and maintaining consistency across the decoupled services inherent in ES/CQRS.

1. **Spec-Driven Blueprint Generation:** Kiro Specs are used to formalize the central domain structure. The core event definitions and the high-level microservice boundaries are formalized in the Spec files.7 Kiro is then tasked with generating the repetitive scaffolding necessary for both App A (command handlers, repositories) and App B (event listeners, read model updaters) based on this single Spec schema.4
2. **Steering Docs for Architectural Enforcement:** The structure.md steering document is critical, enforcing microservice naming conventions, domain boundaries, and the mandatory use of asynchronous, event-driven communication protocols. This use ensures that Kiro generates truly decoupled services, maintaining the architectural resilience required by the Event Sourcing pattern.6
3. **Agent Hooks for Protocol Consistency:** An Agent Hook is configured to trigger whenever a new domain event or command definition file is created.8 The hook's action is to verify that the new file adheres to the established ES protocol (e.g., mandatory fields like eventID and timestamp) and uses the required serialization format. This proactive enforcement guarantees consistency across the two distinct applications derived from the skeleton.9

## Section V: Concept 4: Costume Contest - The Eerie Efficiency UI (The Shadow Log)

### A. The Challenge: Haunting Design That Enhances Function

The Costume Contest category requires a user interface that is "haunting," "polished," and uses spooky design elements that **enhance** the application’s core function.1 Superficial styling is insufficient; the design must contribute meaningfully to usability.1

### B. Proposed Solution: "The Shadow Log" (Secure AI Agent Audit Dashboard)

The Shadow Log is a real-time monitoring and compliance dashboard for tracking the activity of autonomous AI agents and developer-defined hooks within a codebase. The application's function is high-stakes security vigilance, auditing for policy violations, unauthorized tool use, or potential AI hallucinations.

The design strategy leverages the aesthetic of **Eerie Efficiency**, aiming to evoke psychological tension and hyper-vigilance—the unnerving feeling of constant threat detection.

1. **Aesthetics (Digital Brutalism):** The design adopts principles of "Digital Brutalism" or "Stark Minimalism," stripping away decoration in favor of raw, uncompromising function.28
   * **Haunting Element:** The interface uses a stark, high-contrast palette (deep blacks, muted grays, limited neon accents) with rigidly enforced typography. This brutalist style is unforgettable because it rejects comfort and forces the user to confront the raw data streams (code, logs), emphasizing the underlying security threats.28
   * **Polish:** The polish is demonstrated through perfect typographic hierarchy, razor-sharp data alignment, and minimal, highly deliberate micro-animations.
2. **Functional Enhancement:** The restricted color palette is a functional choice. Color is reserved exclusively for critical alerts. When the audit system detects a high-severity compliance violation or security incident, the relevant log entry triggers a subtle, high-contrast red-shift or micro-flicker that is impossible to miss.30 This makes the functional element (threat notification) maximally efficient and directly enhances the app's core purpose (monitoring vigilance) through the intentionally unnerving aesthetic.1

### C. Kiro Implementation Rationale (Focus on Steering for Design Constraints)

Kiro is used to guarantee the rigorous consistency required by the Brutalist design philosophy.

1. **Steering Docs for Visual Constraints:** The product.md steering file is utilized as a rigorous digital design system guardian.5 It specifies non-negotiable constraints, including the exact color hexadecimal codes (e.g., #0A0A0A background), the monospace typeface mandate, and the specific rules for using alert colors.6 When Kiro generates new UI components (e.g., a real-time event log or metric gauge), the Steering Docs force adherence to these restrictive, haunting aesthetic rules, ensuring visual consistency and polish across the entire application.1
2. **Vibe Coding for Polished Components:** Vibe Coding sessions are employed to quickly generate complex, interactive elements, proving Kiro’s ability to produce production-quality, highly responsive code while strictly obeying the non-standard constraints enforced by the Steering Docs.1

## Section VI: Strategic Conclusion and Kiro Utilization Synthesis

The development plan is structured to ensure maximum competitive advantage through deep engagement with Kiro's advanced features and strategic alignment with the prize structure.

### A. Prize Eligibility Maximization

The strategic analysis of prize structure reveals the most profitable combination involves winning one Category Prize and the Best Startup Project Bonus Prize.1

The **Nexus Blueprint (Skeleton Crew)** is the ideal candidate for the Best Startup Project prize ($10,000) because an Event-Sourcing microservice template is inherently a foundational SaaS boilerplate.3 Positioning this submission as the core product of an architecture startup allows the project to potentially win: Best Skeleton Crew ($5,000) AND Best Startup Project ($10,000), resulting in a combined $15,000, plus Post Prizes.1

While the **Phoenix Proxy (Resurrection)** represents the highest technical complexity via its custom MCP server, making it a strong contender for 1st Place Overall ($30,000), the calculated decision involves placing the high-value Startup eligibility on the most architecturally versatile framework (Nexus Blueprint) to ensure a guaranteed high return in the Category + Bonus structure.1

### B. Summary Kiro Utilization Matrix

The following table confirms that all five core Kiro features, including the strategic use of custom MCP, are fully integrated across the portfolio, ensuring high scores in the Implementation criteria.

Table: Kiro Feature Utilization Matrix

| **Kiro Feature** | **Resurrection (Phoenix Proxy)** | **Frankenstein (Chronos Nexus)** | **Skeleton Crew (Nexus Blueprint)** | **Costume Contest (Shadow Log)** |
| --- | --- | --- | --- | --- |
| **Vibe Coding** | Generating legacy API wrappers and mapping logic. | Iterative natural language data querying and trend analysis. | Initial generation of core Event Store services and schemas. | Rapid generation of complex, high-contrast, functional UI components. |
| **Agent Hooks** | Automated test creation/sync for new API endpoints. | Real-time schema validation for incoming purged datasets. | Enforcing microservice communication protocol consistency. | Synchronizing dark-mode tokens across SCSS files (Style Consistency). |
| **Steering Docs** | Defining API standards, error handling, and COBOL language stack (tech.md). | Specifying data governance policies and licensing constraints (product.md). | Enforcing the strict Event Sourcing architecture principles (structure.md). | Hard-coding visual constraints (color palette, typography) (product.md). |
| **Specs/Spec-Driven Dev** | Formalizing COBOL-to-JSON mapping logic in design.md. | Structuring the data ingestion and analysis flow (tasks.md). | Breaking down microservice domain logic and event schema (Requirements -> Design). | Formalizing UI component feature sets and accessibility criteria. |
| **MCP** | **Custom MCP Server:** Ingesting COBOL data dictionary for context-aware code generation (Max Implementation). | **Standard Kiro Tools** (Focus on API/Data Stitching) | **Standard Kiro Tools** (Focus on Foundational Versatility) | **Standard Kiro Tools** (Focus on Steering/Vibe Polish) |

### C. Actionable Recommendation

The recommendation is to proceed immediately with the foundational setup, securing the mandatory compliance elements (public repository, licensing, and the /.kiro directory).1 Development focus must be on the complex context injection strategy, beginning with the custom MCP server implementation for the Phoenix Proxy, as this component is the most significant technical differentiator for maximizing the Implementation score across the entire portfolio.1 The Skeleton Crew project must maintain strict adherence to the two-repository folder requirement to ensure eligibility for the high-value Bonus Prize.1

#### Works cited

1. Kiroween\_hackathon.docx
2. Archivists Work to Identify and Save the Thousands of Datasets Disappearing From Data.gov - 404 Media, accessed November 1, 2025, <https://www.404media.co/archivists-work-to-identify-and-save-the-thousands-of-datasets-disappearing-from-data-gov/>
3. SaaS Boilerplates and Starter Kits, accessed November 1, 2025, <https://www.saaspegasus.com/guides/saas-boilerplates-and-starter-kits/>
4. Kiro Agentic AI IDE: Beyond a Coding Assistant - Full Stack Software Development with Spec Driven AI | AWS re:Post, accessed November 1, 2025, <https://repost.aws/articles/AROjWKtr5RTjy6T2HbFJD_Mw/%F0%9F%91%BB-kiro-agentic-ai-ide-beyond-a-coding-assistant-full-stack-software-development-with-spec-driven-ai>
5. Steering - Docs - Kiro, accessed November 1, 2025, <https://kiro.dev/docs/steering/>
6. Master Kiro Steering Docs in Minutes - AWS Builder Center, accessed November 1, 2025, <https://builder.aws.com/content/32ocJQtMKLT0I8zUp3Kg8C3eAkJ/master-kiro-steering-docs-in-minutes>
7. Specs - Docs - Kiro, accessed November 1, 2025, <https://kiro.dev/docs/specs/>
8. Hooks - Docs - Kiro, accessed November 1, 2025, <https://kiro.dev/docs/hooks/>
9. Automate Your Development Workflow with Kiro's AI Agent Hooks, accessed November 1, 2025, <https://kiro.dev/blog/automate-your-development-workflow-with-agent-hooks/>
10. Model Context Protocol (MCP) - Docs - Kiro, accessed November 1, 2025, <https://kiro.dev/docs/mcp/>
11. Build AI's Future: Model Context Protocol (MCP) with Spring AI in Minutes, accessed November 1, 2025, <https://www.youtube.com/watch?v=MarSC2dFA9g>
12. Building with MCP and the Claude API, accessed November 1, 2025, <https://www.youtube.com/watch?v=aZLr962R6Ag>
13. COBOL Modernization: Benefits, Challenges & 5 Critical Techniques - Swimm, accessed November 1, 2025, <https://swimm.io/learn/cobol/cobol-modernization-benefits-challenges-and-5-critical-techniques>
14. Integrating Legacy Systems with Modern Software: What You Need to Know - MindInventory, accessed November 1, 2025, <https://www.mindinventory.com/blog/how-to-integrate-legacy-system/>
15. COBOL Modernization: Why You Need It + How To Get It Right - OpenLegacy, accessed November 1, 2025, <https://www.openlegacy.com/blog/cobol-modernization>
16. Unlock your development productivity with Kiro and Model Context Protocol (MCP), accessed November 1, 2025, <https://kiro.dev/blog/unlock-your-development-productivity-with-kiro-and-mcp/>
17. More than 90% companies have seen their AI investments failing and blame for this should go to ..., says study - Times of India, accessed November 1, 2025, <https://timesofindia.indiatimes.com/technology/tech-news/more-than-90-companies-have-seen-their-ai-investments-failing-and-blame-for-this-should-go-to-says-study/articleshow/124241294.cms>
18. Thousands of datasets from Data.gov have disappeared since Trump's inauguration. What's going on? | Mashable, accessed November 1, 2025, <https://mashable.com/article/government-datasets-disappear-since-trump-inauguration>
19. Where Our Data Comes From – Niche, accessed November 1, 2025, <https://www.niche.com/about/data/>
20. Top 10 Dataset Websites of 2025 - Medium, accessed November 1, 2025, <https://medium.com/@datajournal/top-10-dataset-websites-of-2024-e81fbb44c18e>
21. Uncovering niche markets using AI - Market Logic, accessed November 1, 2025, <https://marketlogicsoftware.com/blog/uncovering-niche-markets-using-ai/>
22. History of the Capital AI & Market Failures in the Attention Economy - Andrew Kortina, accessed November 1, 2025, <https://kortina.nyc/essays/market-failures-in-the-attention-economy/>
23. Vibe Coding with Kiro, Part I - Medium, accessed November 1, 2025, <https://medium.com/@trobcloud/vibe-coding-with-kiro-part-i-154e0ecbb1fe>
24. AWS Kiro and Vibe Coding: The New Era of AI-Assisted Development, accessed November 1, 2025, <https://builder.aws.com/content/303ZHz2z9euWJXG13EYcpqIaWGX/aws-kiro-and-vibe-coding-the-new-era-of-ai-assisted-development>
25. Event sourcing pattern - AWS Prescriptive Guidance, accessed November 1, 2025, <https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-data-persistence/service-per-team.html>
26. Pattern: Event sourcing - Microservices.io, accessed November 1, 2025, <https://microservices.io/patterns/data/event-sourcing.html>
27. Event Sourcing Explained: The Pros, Cons & Strategic Use Cases for Modern Architects, accessed November 1, 2025, <https://www.baytechconsulting.com/blog/event-sourcing-explained-2025>
28. Comparing UI/UX Design Trends: USA vs Global Agencies, accessed November 1, 2025, <https://design-studio.medium.com/comparing-ui-ux-design-trends-usa-vs-global-agencies-528c93b038a9>
29. Unraveling Horrors: The Scariest UI/UX Design Mistakes That Haunt Users - Skynix LLC, accessed November 1, 2025, <https://skynix.co/resources/unraveling-horrors-the-scariest-ui-ux-design-mistakes-that-haunt-users>
30. Cutting-Edge UI/UX Trends Every Designer Should Embrace - Medium, accessed November 1, 2025, <https://medium.com/design-bootcamp/cutting-edge-ui-ux-trends-every-designer-should-embrace-e71a6660a344>