Memo

To: Emillia Hughes
From: John Ivan Busulwa
Date: December 8, 2025
Subject: Learnings from the Credtics Prototype Development

This memo reflects on the development of the Credtics Credit Management Platform prototype, focusing specifically on the utilization of AI as a building partner.

1. How You Actually Used AI While Building

The Credtics prototype was built using a hybrid approach, where AI served as a **"vibe coding" partner** to accelerate design and configuration, while human judgment remained critical for system architecture and logic validation.

AI Tools and Tasks

The primary tool used was **vibe coding** via generative models, specifically to define and structure the core logical components that would eventually be simulated in the prototype.

- **Task 1: Structured Data Configuration** (Code Generation/Logic Definition): The most critical reliance on AI was for generating the **Data Normalization Prompt** and the **Scoring Rule Generation Prompt**. This defined the expected input/output structure for the mock API calls. I asked the AI to "Generate a Python function that takes raw form data and normalizes it into a clean JSON format" and to "Generate a set of 8 common conventional credit scoring rules". This allowed us to quickly define the structure for the conventional scoring logic and the necessary data cleansing logic for the prototype's JavaScript objects.
- **Task 2: Transparency and Explanation Design** (Prompt Engineering): I used AI to refine the **Justification Text Prompt**, asking it to act as an "Auditing Agent" to ensure the output was concise, cited the data, and avoided financial jargon. This helped establish the tone and function of the AI's most visible user-facing feature.

The Importance of Human Editing and Judgment

While AI rapidly structured the logic, human editing and judgment were indispensable. The initial normalization prompts were too generic, leading to inconsistent output formatting. The human intervention involved:

- **Logic Refinement and Schema Enforcement:** We explicitly iterated the prompt by adding a **strict JSON schema requirement** and instructing the model to output a Python function definition to stabilize the mock output structure significantly.

2. Why the AI Feature in Your Product Looks the Way It Does

The AI feature set in Credtics—Data Normalization, Risk Flag Generation, and Transparent Justification Text—is a deliberate, scoped-down configuration focused on solving the **systemic inefficiency** of protracted loan approval cycles (3-5 days) and high operational costs in microfinance.

Strategic Feature Selection

I chose the current AI features because they target the **initial, manual, and unstructured underwriting bottleneck**.

- **Data Normalization:** Microfinance often involves paper-based or complex, unstructured data. This generative task immediately addresses the input problem by converting messy data (e.g., `"500,000 Shillings"` to `500000.00`) into a clean format that downstream conventional systems can use.
- **Justification Text & Risk Flag:** These features provide an **accurate, transparent credit score and risk recommendation in under 10 minutes**, replacing days of manual analysis. The generative model is leveraged for explanation (**transparency**), which is critical for auditability and compliance, rather than for the core scoring logic itself, which is handled by more reliable, conventional custom rules.

Scoping and Connection to Core Value

For practicality, the scoring algorithm itself was **simplified or scoped down** to conventional, rule-based logic. This was a critical trade-off:

- **Simplification:** The prototype **simulates** the AI calls via hardcoded JSON and uses static JavaScript objects for scoring rules. This limited the ability to demonstrate latency/rate limits and real-time data fetching.
- **Core Value Proposition:** The AI features directly connect to the core value proposition: immediate generation of a clear, auditable risk assessment that cuts the loan approval cycle from 3-5 days to under 10 minutes. The AI provides speed and transparency; the conventional rules provide stability and control.

3. Risks, Trade-offs, and Integrity

Privacy, Data Use, and Security

Since the prototype is a single-page web application that stores all data in JavaScript objects and does not integrate any external APIs, the immediate **security and privacy risks are mitigated** for the prototype stage. However, the production version will require:

- **Data Safety:** Robust data validation checks upon normalization completion.
- **Security:** Integration with MFI Core Banking and Regional Data Providers (Telcos/Utilities) will necessitate rigorous security protocols for handling non-traditional data and loan disbursement/repayment tracking.

Bias and Fairness

I explicitly chose to rely on conventional, rules-based credit scoring logic which can be transparently audited and customized by the MFI client. This choice limits the introduction of opaque, generative-model-derived bias into the core decision engine. The AI's role is primarily to process data and provide explanations, tasks where its outputs are strictly constrained and auditable.

Over-Reliance on AI / User Trust

To prevent **over-reliance**, I implemented two explicit guardrails:

- **Mandatory HIL:** The core function of the system is the **Human-in-the-Loop (HIL) review**. The system provides a *recommendation*, and the Loan Officer (LO) must apply their "local context and judgment" to finalize the decision.

Academic Integrity and Honest Use of AI

In the project's own development, AI was used strictly as a **logic definer and structural co-pilot** (vibe coding), not as a source of final, unedited narrative or design. All code structure, architectural choices, and narrative content in the PRD (like this memo) are the result of human synthesis and validation, with AI acting only to draft the *expected function and structure* of the application's internal components.

4. What You Learned About Building with GenAI

Biggest Surprise or Challenge

The primary challenge was achieving stable, reliable, and standardized output for utility tasks like Data Normalization. Inconsistent initial prompts would "break downstream systems." Surprisingly, for backend utility tasks, a simple, direct prompt proved less effective than a highly constrained and specific one.

One Thing to Teach Another Founder

If I were to teach another founder one thing about using GenAI tools well, it would be: For system integration, treat the AI like a highly constrained API, not a conversational partner.

To ensure the output is immediately usable by downstream systems, you must enforce structure through:

1. **Structured Output:** Use strict JSON Schemas or Pydantic models to enforce output format.
2. **Role-Playing/Persona:** Use role-playing (e.g., "Act as an Auditing Agent") to enforce non-functional constraints like transparency, conciseness, and tone.

Impact on Future Ventures

This project profoundly affects how I think about AI in future ventures. I now see GenAI's highest value not as a replacement for core domain logic (e.g., complex credit scoring), but as an **enabler of system efficiency and transparency**.

Future capstone or ventures will prioritize GenAI for:

- **Input Normalization/Preprocessing:** Using structured output to clean up unstructured data, making it ready for conventional algorithms.
- **Transparent Explanation and Audit Trail:** Using generative text for auditable justification, which is critical for regulated industries.

The key takeaway is that an **AI/Conventional Hybrid Architecture** leveraging AI for speed and explanation, and conventional logic for stability and control.