



DATA | ANALYTICS | AI

Hackathon Concept Label iQ



© 2024 Massive Insights Inc. Proprietary & Confidential

Label iQ – Introduction.

**Prescription drug
labels are designed
for compliance, not
comprehension.**

Let's change that.



© 2024 Massive Insights Inc. Proprietary & Confidential

Label iQ – Introduction

The Challenge

Prescription drug labels are long, technical, and difficult for most readers to understand. Even healthcare professionals struggle to scan 40-page documents filled with complex terminology.

Patient versions such as Medication Guides or PPIs are inconsistent in tone and quality.

The FDA's proposed Patient Medication Information rule aims to improve clarity, but implementation is still years away, leaving no consistent benchmark for readability today.

Our Approach

Label iQ uses **Denodo Platform 9**, the **Denodo AI SDK**, and **Amazon Bedrock** to turn FDA drug labels into an interactive, trustworthy experience.

It powers a natural-language "**Ask the Label**" chat that retrieves relevant text directly from the official FDA label, quotes it verbatim, and summarizes it in clear, plain language with citations and disclaimers.

This approach demonstrates how GenAI can be applied safely and transparently to regulated content.

Why it Matters

- Builds public trust by making complex medical information easier to find, access, and understand.
- Improves comprehension of warnings, dosage, and interactions, helping patients and caregivers use medications more safely.
- Demonstrates responsible use of GenAI and semantic-layer technology within regulated industries.



Label iQ – Executive Summary

1. Concept

Label iQ is a GenAI prototype built with **Denodo Platform 9**, the **Denodo AI SDK**, and **Amazon Bedrock**.

It enables natural-language querying of FDA drug labels. Users can ask questions like "What are the warnings for this drug?" and receive **quote-first, evidence-backed responses** generated directly from official FDA Structured Product Labeling (SPL) text.

The goal is to demonstrate how Denodo's semantic layer and AI SDK can safely power retrieval-augmented generation in a regulated data context.

2. Data Sources

FDA / NIH DailyMed Structured Product Labels (SPL):

25 FDA-approved prescription-drug labels converted to plain text and stored in **Amazon S3**.

Metadata Index:

Each file linked to a label_id, drug_name_display, and snapshot_date in a CSV used as the Denodo base view.

Supporting Dataset (offline):

Readability scores (Flesch, Flesch-Kincaid, SMOG) computed once per label for PR and context, not part of runtime queries.

3. Architecture

Data Layer (Denodo):

- Create a virtual database containing two base views: one for the label index, one for label text.
- Join them into a single derived view (dv_labels) for unified access.
- Enrich schema with semantic descriptions using **Denodo Assistant** (e.g., "label_text: full FDA label body text").

AI Integration:

- The **Denodo AI SDK** connects the semantic layer to **Amazon Bedrock**, using Bedrock's LLM as the reasoning engine.
- Metadata and sample rows are synchronized to **Amazon OpenSearch Serverless (VectorDB)** for semantic retrieval via "Sync VectorDB."
- The Bedrock model uses retrieved text snippets as context to generate structured JSON output: direct quotes, short paraphrase, and citation.

Governance:

- Apply a minimal **Global Security Policy** masking one synthetic field to demonstrate role-based data protection.

4. AI SDK and Chatbot Design

Retrieval and Prompting:

- The AI SDK retrieves the most relevant passages from dv_labels.label_text using embeddings stored in OpenSearch.
- Prompt instructs Bedrock to answer only from these passages, quoting verbatim first, then writing a short plain-language paraphrase.
- If no evidence is retrieved, the model must reply "Not stated in this label."

Response Format:

- Output JSON includes evidence (list of quotes), summary (120–160 words), label_id, drug_name, and a fixed disclaimer.
- Every answer displayed with citations and the line: "Educational only — not medical advice."

Frontend:

- Lightweight web interface with drug dropdown, text input, and structured response panel.
- Calls Denodo AI SDK endpoint and renders quotes, summary, citation, and disclaimer.



Label iQ – One Page Dev Brief

1) Objective

Build a small web app that answers plain-language questions using the official FDA label text for a frozen set of 25 drugs. Answers must quote the label first, then provide a short paraphrase, with visible citation and disclaimer. Separately, produce a one-time readability scoring CSV and single slide. The chatbot must be implemented through the **Denodo AI SDK** integrated with **Amazon Bedrock**.

2) Deliverables

Label iQ Web App

- Single page with drug selector, question input, and answer panel.
- Answer panel shows: verbatim quotes from the label, a concise paraphrase, citation (drug name and label_id), and a fixed disclaimer.
- If no relevant text is found, return "Not stated in this label."

Frozen Label Snapshot

- Folder of 25 label text files and an index mapping ids to drugs and files.

Readability Scores (adjunct, offline)

- clarity_scores.csv and one ranked slide. Not integrated into Denodo.

3) Platforms and services

Denodo Platform 9.x

- Design Studio for base and derived views.
- Denodo Assistant for semantic descriptions.
- Global Security Policies for a simple masking demo.
- Denodo AI SDK for chatbot integration.

Amazon Web Services

- Amazon S3 for the static snapshot.
- Amazon Bedrock foundation model as the LLM endpoint used by the AI SDK.
- Amazon OpenSearch Serverless (vector store) for the AI SDK's "Sync VectorDB" step.

Frontend

- Minimal web UI (framework of choice).

4) Data inputs and layout

Convert each FDA label to plain UTF-8 text. One canonical file per drug.

File layout:

```
/snapshot/  
labels/{label_id}.txt  
label_index.csv # columns: label_id, drug_name_display, file_path, snapshot_date  
No runtime calls to external data sources.
```

5) Denodo objects and naming

Create a new Virtual Database for Label iQ.

A. Base Views (Design Studio → Create Base Views from files on S3)

dv_label_index over label_index.csv
dv_label_text over the folder of label text files, exposing two fields:

- label_id (string)
- label_text (large text)

B. Derived Views (Design Studio → New → Selection/Join as needed)

dv_labels joining by label_index to bv_label_text on label_id

Fields: label_id, drug_name_display, snapshot_date, label_text

C. Semantic enrichment (Denodo Assistant)

Add descriptions at view and field level, for example:

- dv_labels.label_text: "Full FDA label body text for the selected drug. Use as the only evidence source for answers."
- dv_labels.label_id: "Stable id used in citations."

D. Tagging and catalog

Tag dv_labels with simple categories like label_text, citation, source_snapshot to help the AI SDK. Optional: publish to the Denodo Data Catalog for exploration.

E. Global Security Policy (simple governance demo)

Create a small synthetic helper view, e.g., bv_contacts with an email column, apply a column-masking policy for the "analyst" role. Do not expose this in the UI; it is only to show policy capability.

6) AI SDK configuration

A. Vector store sync

Run the AI SDK's **Sync VectorDB** against dv_labels, embedding only label_text, and include metadata fields label_id, drug_name_display, snapshot_date.

B. Chatbot profile instructions

Create a profile with these enforced instructions:

- Answer only from text retrieved by the AI SDK from dv_labels.label_text.
- Return quotes first, then a short paraphrase. Do not introduce facts not present in the quotes.
- If no relevant text is retrieved, return exactly: "Not stated in this label."
- Append the disclaimer: "Educational only, not medical advice."

C. Routing

Start with simple keyword intent hints in the profile (dosage, warning, interactions, pregnancy), but allow the retriever to search for the full label_text. No per-section routing is required for MVP.

D. Output contract

Configure the AI SDK to return a JSON object with:

- drug_name (string)
- label_id (string)
- evidence (array of {quote_start_char, end_char})
- summary (string, 120–160 words)
- disclaimer (string)

Reject any response with empty evidence unless it is exactly the fallback line.

7) Web app behavior

Drug dropdown populated from dv_labels via a simple API endpoint.

Question textbox and Submit button.

On submit:

- Call the AI SDK chatbot endpoint with label_id and question.
- Render each evidence quote in order, then the summary, then citation "Source: FDA label, (drug_name), (label_id)", then the disclaimer.

If the fallback string is returned, display it plainly.

8) Prompt content used by the AI SDK

System prompt (profile):

"You are Label iQ. You answer only from FDA label text supplied as retrieved snippets. Quote the relevant sentences first, then write a concise paraphrase in plain language using only the quoted information. Do not provide medical advice. If the provided text does not contain an answer, return 'Not stated in this label.' Include the disclaimer verbatim."

Model input template passed by the SDK:

Drug name

Label id

User question

Retrieved snippets list

Model output format:

The JSON structure defined in section 6D.

9) Readability scores (adjunct, offline)

Run a separate script outside Denodo over each labels/{label_id}.txt to compute Flesch Reading Ease, Flesch-Kincaid Grade Level, and SMOG.

Produce clarity_scores.csv with:

label_id, drug_name_display, fre, fkgl, smog, composite_0_100, word_count, snapshot_date

Produce one simple ranked slide. Do not ingest these into Denodo.

10) Testing plan

Gold test set: at least 15 questions that cover dosage, boxed warning, interactions, use in specific populations, storage conditions.

For each of the 25 labels, run 5 random questions. Verify:

- Evidence quotes are exact substrings of label_text.
- Summary length is within 120–160 words and uses no unquoted facts.
- Fallback is returned when the label does not address the question.
- Citation line and disclaimer are always present.

11) Acceptance criteria

Chat returns citation-backed answers for all gold questions with zero uncited claims.

Fallback is returned correctly when appropriate.

No calls are made to external data sources at runtime.

clarity_scores.csv and the ranked slide are delivered and match the snapshot.

This spec reflects the hackathon manual's flow: base and derived views in Denodo, semantic enrichment via Denodo Assistant, VectorDB sync, profile instructions in the AI SDK, and Amazon Bedrock as the LLM.

