☰ STEP 3: AI-POWERED DASHBOARD REPORT

Talent Match Intelligence Dashboard – Case Study 2025

Executive Summary

The **Talent Intelligence Dashboard** serves as the operational interface for the Success Formula (Step 1) and SQL Matching Engine (Step 2). It transforms complex data analytics into an intuitive, decision-support tool for HR managers and business leaders.

By integrating **Real-time Talent Matching** with **Generative AI**, the dashboard enables users to: 1. **Identify Top Talent:** Instantly match employees against high-performer benchmarks using the validated Success Formula. 2. **Generate Job Profiles:** Create standardized, competency-aligned job descriptions using AI, which then serve as immediate inputs for talent searching. 3. **Visualize Gaps:** Understand *why* a candidate is a match through detailed gap analysis visualizations.

This report documents the system architecture, feature logic, and technical implementation of the dashboard application.

---

1. System Architecture

The application follows a modern, layered architecture designed for performance and scalability.

1.1 High-Level Design

Frontend Layer: Built with **Streamlit** (Python), offering a responsive and interactive user interface.

Service Layer: Python modules (`core/matching.py`, `core/job_generator.py`) handle business logic and state management.

Data Layer: PostgreSQL (Supabase) executes the heavy-lifting via the 18-stage SQL pipeline defined in Step 2.

AI Layer: Google Gemini provides generative capabilities for job profile creation and refinement.

1.2 Data Flow Diagram

```
graph TD
    User[User] -->|Interacts| UI[Streamlit Dashboard]
    UI -->|Request| Logic[Python Service Layer]

    subgraph "Talent Matching"
    Logic -->|Parameters| SQL[SQL Engine (Step 2)]
    SQL -->|Query| DB[(Supabase DB)]
    DB -->|Result Set| SQL
    SQL -->|DataFrame| Logic
    end

    subgraph "AI Job Gen"
    Logic -->|Prompt| AI[Google Gemini API]
    AI -->|JSON Output| Logic
    Logic -->|Save| DB
    end

    Logic -->|Render| UI
```

1.3 Project Dependencies (The Trilogy)

The dashboard is the final piece of a three-part ecosystem, relying heavily on the foundations built in previous steps:

```
graph LR
    S1[Step 1: Success Pattern] -->|Formula & Weights| S2[Step 2: SQL Engine]
    S2 -->|Logic & Pipeline| S3[Step 3: Dashboard]
```

Dependency on Step 1: The dashboard's scoring logic (50/25/10/10/5) and competency definitions are directly derived from the empirical Success Patterns discovered in Step 1.

Dependency on Step 2: The dashboard does not calculate scores itself. It acts as a client for the 18-stage SQL pipeline built in Step 2, ensuring 100% consistency between backend logic and frontend presentation.

1.4 Feature Summary

Module	Purpose	Key Interaction	Output
Talent Matching	Compare employees to benchmark	Filters, Mode Selection	Ranking + Gap Analysis
AI Job Generator	Generate job profiles	Role Input + Revision	JSON Structured Profile
Auto-Match	Use AI output to search	One-click "Find Talent"	Matched Candidates
Breakdown View	Visualize competency gaps	Drill-down click	TV & TGV Bar Charts

2. Feature Deep Dive: Talent Matching Engine

The core of the dashboard is the Talent Matching Engine, which operationalizes the Success Formula using the **50/25/10/10/5** weighted model established in our analysis.

2.1 Dual-Mode Search Strategy

To support diverse HR use cases, the engine operates in two distinct modes:

Mode A: Manual Benchmark (Succession & Cloning)

Concept: "Find me people who are like Employee X."

Workflow: User selects one or more high-performing employees. The system calculates their median attributes to form a dynamic baseline.

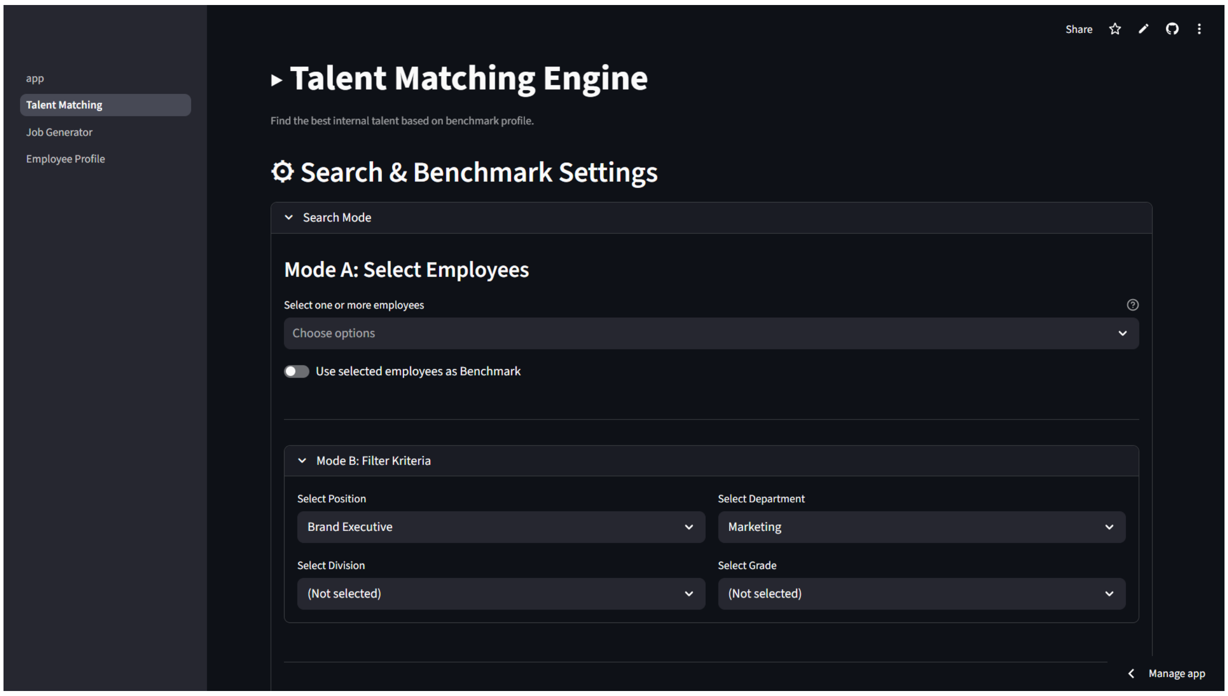
Use Case: Identifying successors for a specific leader or finding "clones" of a top sales performer.

Mode B: Filter Benchmark (Role-Based Matching)

Concept: "Find me the best fit for this specific role/grade/department."

Workflow: User applies filters (e.g., "Senior Managers in Marketing"). The system identifies all High Performers (Rating=5) matching these criteria to build the baseline.

Use Case: Standard recruitment for open vacancies.



▸ Talent Matching Engine

Find the best internal talent based on benchmark profile.

⚙ Search & Benchmark Settings

Search Mode

Mode A: Select Employees

Select one or more employees: Choose options

Use selected employees as Benchmark

Mode B: Filter Kriteria

Select Position: Brand Executive

Select Department: Marketing

Select Division: (Not selected)

Select Grade: (Not selected)

Figure 1: Talent Matching Interface showing Mode selection and filters.

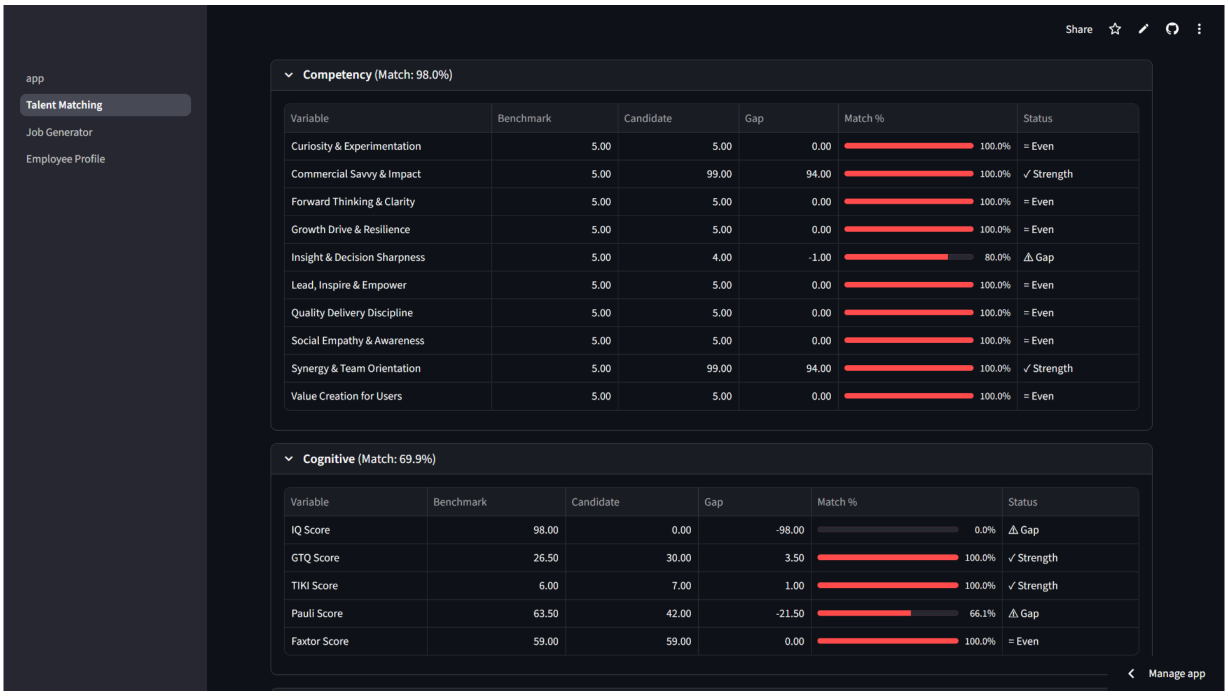
2.2 Visualization & Output

The dashboard prioritizes clarity and decision speed:

The Podium: The top 3 candidates are highlighted in a "Winner's Podium" layout, focusing attention on the best matches.

Detailed Gap Analysis: Users can drill down into any candidate to see a detailed breakdown.

- **Bar Charts** visualize the candidate's score vs. the benchmark baseline for every Talent Variable.
- This answers the critical "Why?" question behind every match score.



Competency (Match: 98.0%)

Variable	Benchmark	Candidate	Gap	Match %	Status
Curiosity & Experimentation	5.00	5.00	0.00	100.0%	= Even
Commercial Savvy & Impact	5.00	99.00	94.00	100.0%	✓ Strength
Forward Thinking & Clarity	5.00	5.00	0.00	100.0%	= Even
Growth Drive & Resilience	5.00	5.00	0.00	100.0%	= Even
Insight & Decision Sharpness	5.00	4.00	-1.00	80.0%	⚠ Gap
Lead, Inspire & Empower	5.00	5.00	0.00	100.0%	= Even
Quality Delivery Discipline	5.00	5.00	0.00	100.0%	= Even
Social Empathy & Awareness	5.00	5.00	0.00	100.0%	= Even
Synergy & Team Orientation	5.00	99.00	94.00	100.0%	✓ Strength
Value Creation for Users	5.00	5.00	0.00	100.0%	= Even

Cognitive (Match: 69.9%)

Variable	Benchmark	Candidate	Gap	Match %	Status
IQ Score	98.00	0.00	-98.00	0.0%	⚠ Gap
GTQ Score	26.50	30.00	3.50	100.0%	✓ Strength
TIKI Score	6.00	7.00	1.00	100.0%	✓ Strength
Pauli Score	63.50	42.00	-21.50	66.1%	⚠ Gap
Faxtor Score	59.00	59.00	0.00	100.0%	= Even

Figure 2: Detailed Gap Analysis showing candidate scores against the benchmark baseline.

2.3 Scoring Logic (Target State)

The dashboard displays scores based on the finalized Success Formula weights: * **Core Competencies:** 50% * **Work Style (PAPI):** 25% * **Cognitive Ability:** 10% * **Strengths:** 10% * **Personality:** 5%

---

3. Feature Deep Dive: AI Job Profile Generator

This module leverages Large Language Models (LLM) to streamline the recruitment process.

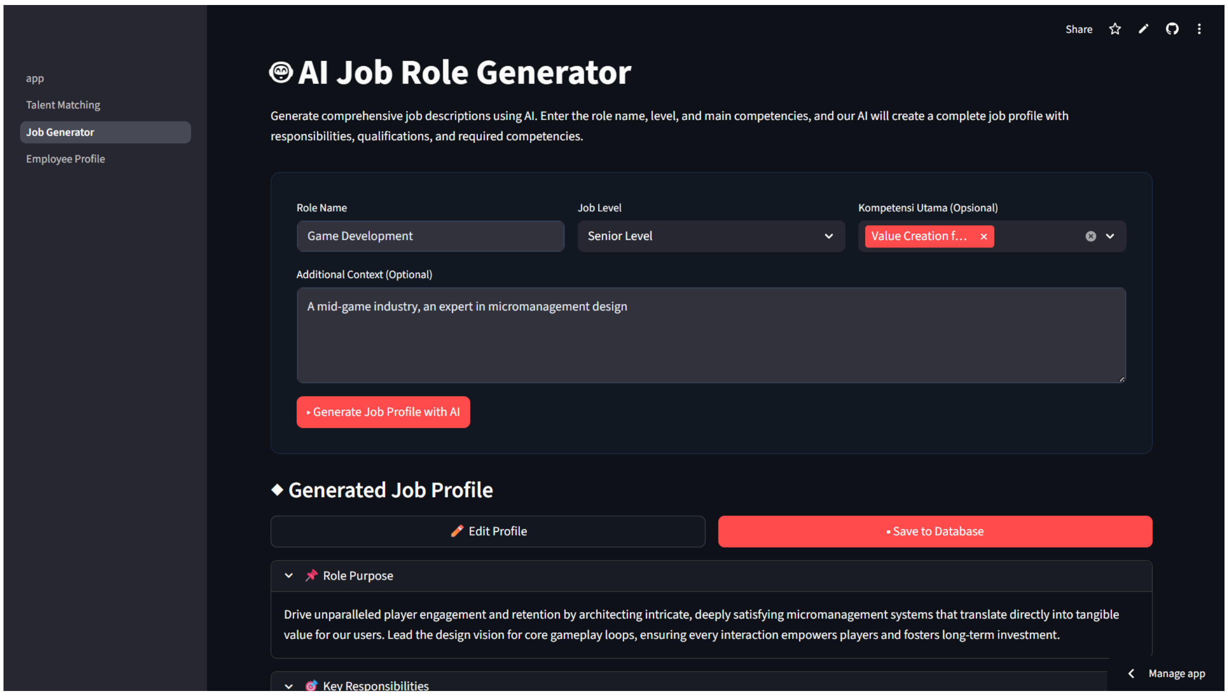
3.1 The "Workflow Bridge"

A key innovation is the seamless integration between job creation and talent searching:

Generate: User inputs a role name (e.g., "Product Manager") and optional context. AI generates a full profile.

Refine: User can request AI revisions (e.g., "Make it more senior") or manually edit the text.

Save & Match: Once saved to the database, a **"Find Matching Talents"** button appears. Clicking this immediately triggers the Talent Matching Engine (Mode B) using the new role's criteria as the filter.



AI Job Role Generator

Generate comprehensive job descriptions using AI. Enter the role name, level, and main competencies, and our AI will create a complete job profile with responsibilities, qualifications, and required competencies.

Role Name: Game Development

Job Level: Senior Level

Kompetensi Utama (Opsional): Value Creation f...

Additional Context (Optional): A mid-game industry, an expert in micromanagement design

Generate Job Profile with AI

Generated Job Profile

Edit Profile | Save to Database

Role Purpose

Drive unparalleled player engagement and retention by architecting intricate, deeply satisfying micromanagement systems that translate directly into tangible value for our users. Lead the design vision for core gameplay loops, ensuring every interaction empowers players and fosters long-term investment.

Key Responsibilities

Figure 3: AI Job Generator showing the input form and generated content.

3.2 Prompt Engineering Strategy

To ensure professional and usable outputs, we employ advanced prompt engineering techniques:

Structured JSON Output: The AI is instructed to return data in a strict JSON schema, not free text. This allows the application to parse responsibilities, qualifications, and competencies into separate UI components.

Competency Mapping: The prompt context includes the organization's specific competency pillars, ensuring generated requirements align with the internal framework.

Privacy First: No employee PII (Personally Identifiable Information) is ever sent to the AI. The model only receives role definitions and generic competency labels.

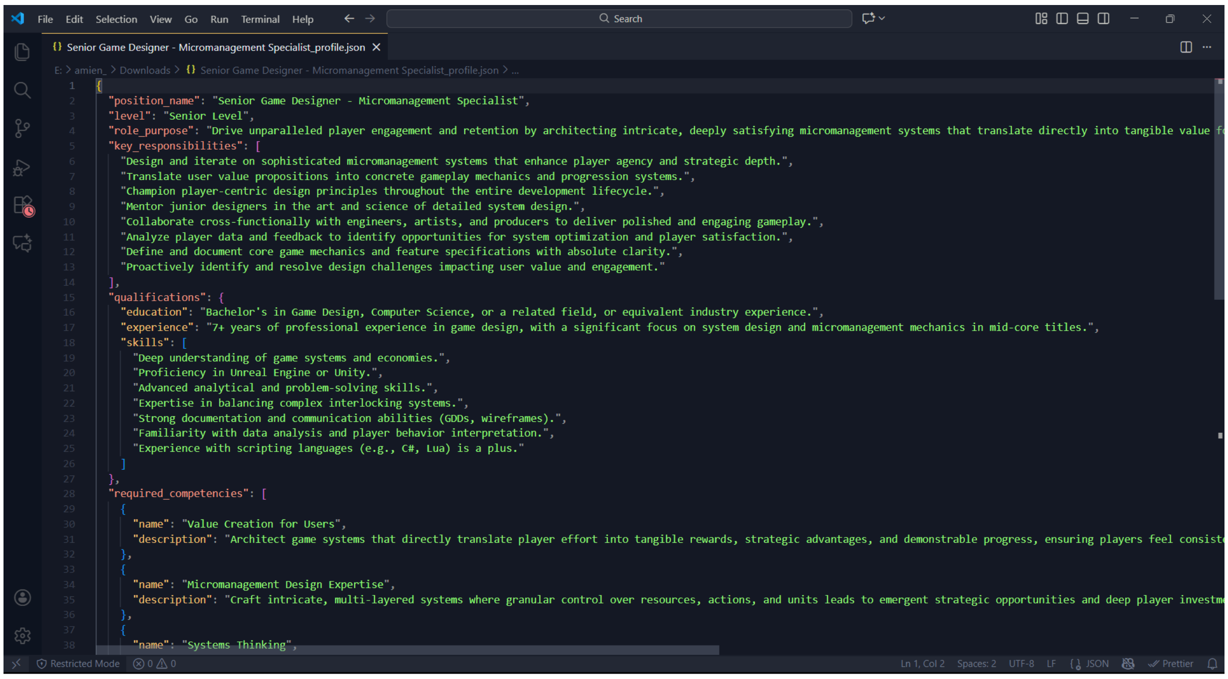
3.3 Output Structure

The AI generates a comprehensive profile including: * **Role Purpose:** A strategic summary. * **Key Responsibilities:**

Action-oriented bullet points. * **Required Competencies:** Mapped to the 10 organizational pillars with context-specific application. * **Qualifications:** Education, experience, and hard skills.

Example JSON Output:

```json
{
  "position_name": "Senior Product Manager",
  "level": "Senior Level",
  "role_purpose": "Lead the product strategy and execution...",
  "key_responsibilities": [
    "Define product vision and roadmap",
    "Collaborate with engineering and design teams"
  ],
  "required_competencies": [
    {
      "name": "Strategic Thinking",
      "description": "Ability to translate market trends into product strategy."
    }
  ],
  "qualifications": {
    "education": "Bachelor's degree in CS or Business",
    "experience": "5+ years in product management",
    "skills": ["Agile", "SQL", "User Research"]
  }
}
```



Senior Game Designer - Micromanagement Specialist_profile.json

```json
{
  "position_name": "Senior Game Designer - Micromanagement Specialist",
  "level": "Senior Level",
  "role_purpose": "Drive unparalleled player engagement and retention by architecting intricate, deeply satisfying micromanagement systems that translate directly into tangible value f
  "key_responsibilities": [
    "Design and iterate on sophisticated micromanagement systems that enhance player agency and strategic depth.",
    "Translate user value propositions into concrete gameplay mechanics and progression systems.",
    "Champion player-centric design principles throughout the entire development lifecycle.",
    "Mentor junior designers in the art and science of detailed system design.",
    "Collaborate cross-functionally with engineers, artists, and producers to deliver polished and engaging gameplay.",
    "Analyze player data and feedback to identify opportunities for system optimization and player satisfaction.",
    "Define and document core game mechanics and feature specifications with absolute clarity.",
    "Proactively identify and resolve design challenges impacting user value and engagement."
  ],
  "qualifications": {
    "education": "Bachelor's in Game Design, Computer Science, or a related field, or equivalent industry experience.",
    "experience": "7+ years of professional experience in game design, with a significant focus on system design and micromanagement mechanics in mid-core titles.",
    "skills": [
      "Deep understanding of game systems and economies.",
      "Proficiency in Unreal Engine or Unity.",
      "Advanced analytical and problem-solving skills.",
      "Expertise in balancing complex interlocking systems.",
      "Strong documentation and communication abilities (GDDs, wireframes).",
      "Familiarity with data analysis and player behavior interpretation.",
      "Experience with scripting languages (e.g., C#, Lua) is a plus."
    ]
  },
  "required_competencies": [
    {
      "name": "Value Creation for Users",
      "description": "Architect game systems that directly translate player effort into tangible rewards, strategic advantages, and demonstrable progress, ensuring players feel consiste
    },
    {
      "name": "Micromanagement Design Expertise",
      "description": "Craft intricate, multi-layered systems where granular control over resources, actions, and units leads to emergent strategic opportunities and deep player investm
    },
    {
      "name": "Systems Thinking",
```

Figure 4: Structured JSON output generated by the AI model.

4. Technical Implementation Details

4.1 State Management

Streamlit is inherently stateless (re-runs the script on every interaction). To create a cohesive multi-step application, we utilize `st.session_state` extensively:

Persistence: Storing generated job profiles so they don't disappear during manual edits.

Navigation: Passing `vacancy_id` and filter parameters from the Job Generator page to the Talent Matching page to enable the "Auto-Match" workflow.

Pagination: Managing current page state for the matching results table.

4.2 Performance Optimization

Caching: We use `@st.cache_data` to load static dimension tables (Departments, Positions, Grades) only once. This ensures sub-second page loads.

SQL Pushdown: All heavy calculations (percentiles, aggregations, weighted sums) are executed in the PostgreSQL engine (Step 2). Python only handles UI rendering, keeping the dashboard responsive even with large datasets.

4.3 Security

Secrets Management: API keys (Gemini) and Database Credentials are stored in `.streamlit/secrets.toml`, ensuring they are never hardcoded in the repository.

Input Validation: All user inputs are sanitized before being passed to SQL queries to prevent injection attacks.

---

5. User Guide (Walkthrough)

Scenario 1: Finding a Successor (Mode A)

Navigate to **Talent Matching**.

In "Search Mode", select **Mode A**.

Search and select the current incumbent (e.g., "John Doe").

Toggle **"Use selected employees as Benchmark"**.

Click **Run Talent Match**.

Review the **Podium** for the top 3 successors and use **Detailed Breakdown** to analyze gaps.

Scenario 2: New Role Recruitment (AI + Mode B)

Navigate to **AI Job Generator**.

Enter "Senior Data Scientist" and select "Senior Level".

Click **Generate Job Profile**.

Review the AI output. Use **"Request Revisions"** if needed (e.g., "Add Python requirement").

Click **Save to Database**.

Click the newly appeared **"Find Matching Talents"** button.

The system automatically switches to the Matching page and finds internal candidates fitting the new role.

---

6. Future Roadmap

AI Interviewer: Generate tailored interview questions based on the specific competency gaps identified in the matching report.

Career Pathing: Visualize potential career trajectories for employees based on their match rates across the entire position hierarchy.

Spider Charts: Enhance the gap analysis visualization with radar charts for multi-dimensional comparison.

---

7. Limitations & Considerations

While the dashboard provides powerful insights, it is important to acknowledge certain limitations:

AI Hallucinations: Although rare with structured JSON prompts, Generative AI can occasionally produce generic or slightly inaccurate requirements. Human review is always recommended before saving.

Data Quality Dependency: The matching engine's accuracy is directly tied to the quality of input data (performance ratings, competency scores). Garbage in, garbage out.

Self-Reported Data: Psychometric data (MBTI, DISC) is self-reported and should be treated as a behavioral indicator rather than an absolute truth.

Performance at Scale: While optimized with CTEs, the current SQL implementation may require additional indexing or materialized views if the employee base grows beyond 10,000 users.

---

8. Conclusion

The **Talent Intelligence Dashboard** successfully completes the analytics loop. It takes the empirical **Success Patterns (Step 1)** and the robust **SQL Logic (Step 2)** and delivers them through a **user-centric AI application**.

By combining rigorous data science with the creative power of Generative AI, the dashboard empowers HR leaders to make faster, fairer, and more data-driven talent decisions.

Report Prepared By: Engki Nandatama (2025)
Date: November 25, 2025
For: Talent Intelligence Dashboard - Case Study Brief rev 1.1
Deliverable: Step 3 - Dashboard Documentation