

Tiber IDP Lab Module: Final Handoff Package

To: J

From: Manus (in collaboration with Claude)

Date: Feb 17, 2026

1. Project Vision & Goal

The **Tiber IDP Lab** is a specialized analytics module designed to provide a comprehensive, data-driven platform for **Individual Defensive Player (IDP)** analysis. While the core Tiber engine (FORGE) focuses on offensive scoring, the IDP Lab fills a critical gap by applying advanced statistical modeling to defensive players. The primary goal is to move beyond simple box-score stats and offer deeper insights into a player's disruptive impact and overall "Defensive DNA."

The module is built on a **modular, scalable architecture** that remains strictly isolated from the offensive scoring engine. This ensures that defensive analytics can evolve independently while still sharing the same high-level design patterns and UI/UX principles as the rest of the Tiber platform.

2. Core Features & Functionalities

The IDP Lab introduces several key features that empower users to evaluate defensive players with unprecedented granularity.

Feature	Description
Havoc Index	A weighted, normalized, and Bayesian-smoothed metric (0-100 scale) that measures a player's disruptive impact per snap.
IDP Rankings	A filterable, sortable leaderboard for all defensive players, with position-specific baselines and Tier T1-T5 badges.
Defensive DNA	Vector-based player trait profiles (e.g., "Power Rusher," "Ball Hawk") for playstyle classification and similarity search.
Player Profile	A detailed view of a player's seasonal stats, weekly performance logs, and a Havoc component breakdown.

Comparison Tool	Side-by-side analysis of two players, featuring a "DNA Radar Chart" overlay for visual playstyle comparison.
AI-Ready Export	A CSV export feature with custom prompt hints, designed for seamless integration with external AI analysis tools.

3. Technical Stack & Architecture

The IDP Lab follows Tiber's existing **Bronze/Silver/Gold** data pattern, ensuring consistency and reliability across the ETL pipeline.

- **Frontend:** Built with **React 18**, **Tailwind CSS**, and **shadcn/ui**. Data fetching is handled by **TanStack Query**, and visualizations use **Chart.js** and **D3.js** (specifically for the Radar charts).
- **Backend:** A **Node.js/TypeScript** API for serving rankings and player details, complemented by a **Python** ETL pipeline for data ingestion and enrichment.
- **Database:** **PostgreSQL** with **Drizzle ORM**. The system leverages the `pgvector` extension for high-performance similarity searches based on trait vectors.
- **Architecture:** The module is designed to be **loosely coupled**, with its own set of routes (`/api/idp/*`) and a dedicated schema file to prevent interference with the FORGE engine.

4. Data Schema & Models

The proposed database schema for the IDP Lab is structured to handle both raw weekly stats and aggregated seasonal insights.

Table Name	Purpose	Key Fields
<code>idp_player_week</code>	Stores raw and enriched weekly performance data.	<code>gsis_id</code> , <code>season</code> , <code>week</code> , <code>defense_snaps</code> , <code>havoc_raw</code>
<code>idp_player_season</code>	Aggregated seasonal stats and final Havoc Index scores.	<code>gsis_id</code> , <code>season</code> , <code>havoc_index</code> , <code>havoc_tier</code> , <code>games</code>
<code>idp_trait_vectors</code>	Stores 12-dimensional trait vectors for Defensive DNA.	<code>gsis_id</code> , <code>season</code> , <code>trait_vector</code> (pgvector) , <code>cluster_label</code>

<code>idp_position_baselines</code>	Per-position-group constants for Z-score normalization.	<code>season</code> , <code>position_group</code> , <code>metric_name</code> , <code>mean_value</code> , <code>std_dev</code>
-------------------------------------	---	---

5. Implementation Roadmap (MVP)

The initial build-out of the IDP Lab is estimated at **25-35 hours of effort**. The roadmap is prioritized to deliver a functional MVP before moving into advanced Phase 2 features.

- 1. Schema & Migrations:** Define the four core IDP tables in a separate Drizzle schema file and apply migrations.
- 2. ETL Pipeline:** Implement the Python-based Bronze (download) and Silver (enrichment) stages, followed by the Node-based Gold (aggregation and scoring) stage.
- 3. API Development:** Create the `/api/idp/rankings` , `/api/idp/player/:id` , and `/api/idp/export/csv` endpoints.
- 4. Frontend Implementation:** Build the main IDP Lab page shell, the filterable rankings table, and the initial player profile view.
- 5. Validation & Testing:** Conduct E2E testing and data validation using the Havoc Index sanity checks (e.g., ensuring a mean of 50 within each position group).

Phase 2 (Post-MVP):

- Implementation of **Defensive DNA** vectors and radar charts.
- **Similar Players API** using KNN search on trait vectors.
- Advanced **Bayesian Havoc models** and scheme intelligence integration.

Attachments:

- `tiber_idp_lab_spec.pdf` (Detailed Technical Specification)
- `tiber_idp_lab_spec_summary.txt` (Key Findings & Formulas)
- `reports/tiber_idp_lab_handoff_package.md` (Initial Brainstorming Draft)