**Project Design Documentation Template:**

Last updated: August 12, 2025

1. Project Title & Version Control  
  
ScoutConnect – Smart Scouting Platform for Talent Discovery & Collaboration  
Version: DRAFT  
Date: 08/12/2025  
Change Log: N/A

Give your project a clear and relevant name (e.g., “Secure Login API with Token Auth” or “Log Analyzer for Failed SSH Attempts”)

*Version Control*

Version: DRAFT

Date: MM/DD/YYYY

Change Log: N/A

2. Project Summary  
  
ScoutConnect is a backend system designed to help coaches and scouts discover and evaluate under-the-radar athletes. The platform centralizes player data, provides intelligent evaluation tools, and fosters collaboration between sports professionals. It matters because it increases the visibility of hidden talent and improves decision-making in athlete recruitment.

What is your project about and why does it matter?

# 3. Problem Statement / Use Case Currently, many talented athletes go unnoticed due to lack of exposure or inconsistent evaluation criteria. ScoutConnect solves this by providing a standardized, data-driven backend platform for storing, analyzing, and sharing athlete performance information. The primary users will be coaches, scouts, and sports organizations looking to identify and track athletic talent.

What real-world problem does this solve? Who would use this?

# 4. Goals and Objectives - Develop a robust backend API with authentication, CRUD operations, and role-based access control. - Implement intelligent scoring and recruitment logic based on universal and sport-specific criteria. - Integrate collaboration tools such as watchlists, comments, and evaluation sharing.

List 2–3 clear and specific technical goals of your project.

# 5. Key Features / Functions - User authentication with JWT and role-based access. - Player profiles with performance evaluations and sport-specific metrics. - Scoring engine with clutch performance and growth tracking. - Recruitability and hidden talent identification. - Player comparison and watchlist system. - Collaboration features: comments, evaluation sharing, tagging. - Stat ingestion from CSV/JSON with automated updates. - Injury risk and recovery tracking.

What are the main things your project will do?

# 6. Tech Stack and Tools - Languages: Python - Framework: FastAPI or Django - Database: PostgreSQL - Containerization: Docker, Docker Compose - Authentication: JWT - Tools: pgAdmin, Swagger/OpenAPI, GitHub, pytest/Django TestCase

Languages, frameworks, platforms, or security tools you'll use (e.g., Python, Flask, PostgreSQL, Zeek, Wireshark)

# 7. Architecture / Workflow Diagram

Algorithm and Flowchart for project

# 8. Timeline / Weekly Milestones Week 1: Project setup: GitHub repo, environment, PostgreSQL, Docker, README. Week 2: Database schema creation (Users, Players, Evaluations, etc.) and authentication. Week 3: Core API for player profiles and evaluations. Week 4: Multi-sport criteria system with reusable templates. Week 5: Scoring engine for universal and sport-specific metrics. Week 6: Recruitability and hidden talent logic. Week 7: Player comparison and watchlist system. Week 8: Collaboration tools: comments, sharing, tagging. Week 9: Stat ingestion and real-time updates. Week 10: Injury tracking and risk prediction. Week 11: Testing, validation, error handling, API docs. Week 12: Final polish, documentation, and demo prep.

# 9. Risks and Risk Mitigation - Complex scoring algorithms could delay development — mitigate by building in stages and testing early. - Data model changes mid-project — mitigate by thorough initial database design and using migrations. - Time constraints — mitigate with strict adherence to weekly milestones.

What risks are there to successful completion of the project and what can you implement to mitigate the impact of those risks.

# 10. Evaluation Criteria - All planned endpoints functional and tested. - Database correctly stores and retrieves all required data. - API documented with Swagger/OpenAPI. - Scoring and recruitability algorithms produce accurate, reproducible results.

How will you know this is done well? List at least 3 measures of success.

# 11. Future Considerations - Integration with external sports data APIs for live updates. - Mobile app front-end integration. - Machine learning model to predict player potential. - Enhanced security with 2FA and advanced logging.

What long term maintenance or added functionality will be needed in the future