PRESENTATION

FOOTBALL
DATA INTELLIGENCE

complete Football Data Intelligence Platform — combining data management, web engineering, and artificial intelligence.

PROJECT STEPS

1. PROJECT MANAGMENT

Jira Service Management

GitHub workflow

Notion

PT

Professionalism & Teamwork

Communication • Collaboration • Time Management • Presentation Quality

Communication

Clear channels (Slack/Email) + documented decisions (Notion) ensure fast alignment and fewer misunderstandings. Use structured standups and summaries.

Collaboration

Pair programming, code reviews, and shared ownership across modules (frontend, backend, ML) increase code quality and speed up problem solving.

Time Management

Kanban boards + sprint goals + time-boxed tasks keep milestones predictable. Prioritize high-impact items and avoid scope creep with short feedback loops.

Presentation Quality

Story-driven demos, crisp slides, and rehearsed pitches communicate technical value and human impact. Use visuals and one-minute summaries for stakeholders.

Team Health — Quick Metrics

95%
Communication Clarity

88%
Collaboration Efficiency

92%
Deadline Respect

94%
Presentation Impact

Recent Highlights

- Delivered demo to stakeholders — high engagement
- Zero critical issues in last sprint
- Peer review coverage: 100%

Celebrate Wins

Updated: **Nov 9, 2025**

Pour vous
Récents
Favoris
Apps
Plans
Espaces
Récents
tech fit project
Plus d'espaces
Équipes
Plus
Envoyer du feedback s...

Espaces

tech fit project

Résumé | Liste | Tableau | Calendrier | Chronologie | Pages | Formulaires

Recherche dans... | CN | Filtrer | Regrouper

IDEA 1

complete Football Data Intelligence Platform — combining data management, web engineering, and artificial intelligence.

KAN-1

Créer

TO DO 10

Design website layout and page structure (Home, Dashboard, Login, Reports)

KAN-3

Define data model for player, match, and training entities

KAN-4

Set up Supabase / PostgreSQL database

KAN-5

Configure ETL pipeline architecture

KAN-6

Plan Keycloak authentication

IN PROGRESS 5

Implement user authentication via Keycloak

KAN-13

Develop ETL scripts to fetch and clean match data

KAN-14

Build player performance dashboard (charts + tables)

KAN-15

Integrate backend API with frontend

KAN-16

Add visualization for training vs

IN REVIEW 4

Data accuracy validation for ETL pipeline

KAN-18

API endpoint testing (Postman or automated tests)

KAN-19

Frontend UI/UX review

KAN-20

PDF report generator test results

KAN-21

DONE 4

Database schema finalized and deployed

~~KAN-22~~

ETL pipeline stable and integrated

~~KAN-23~~

Frontend and backend communication verified

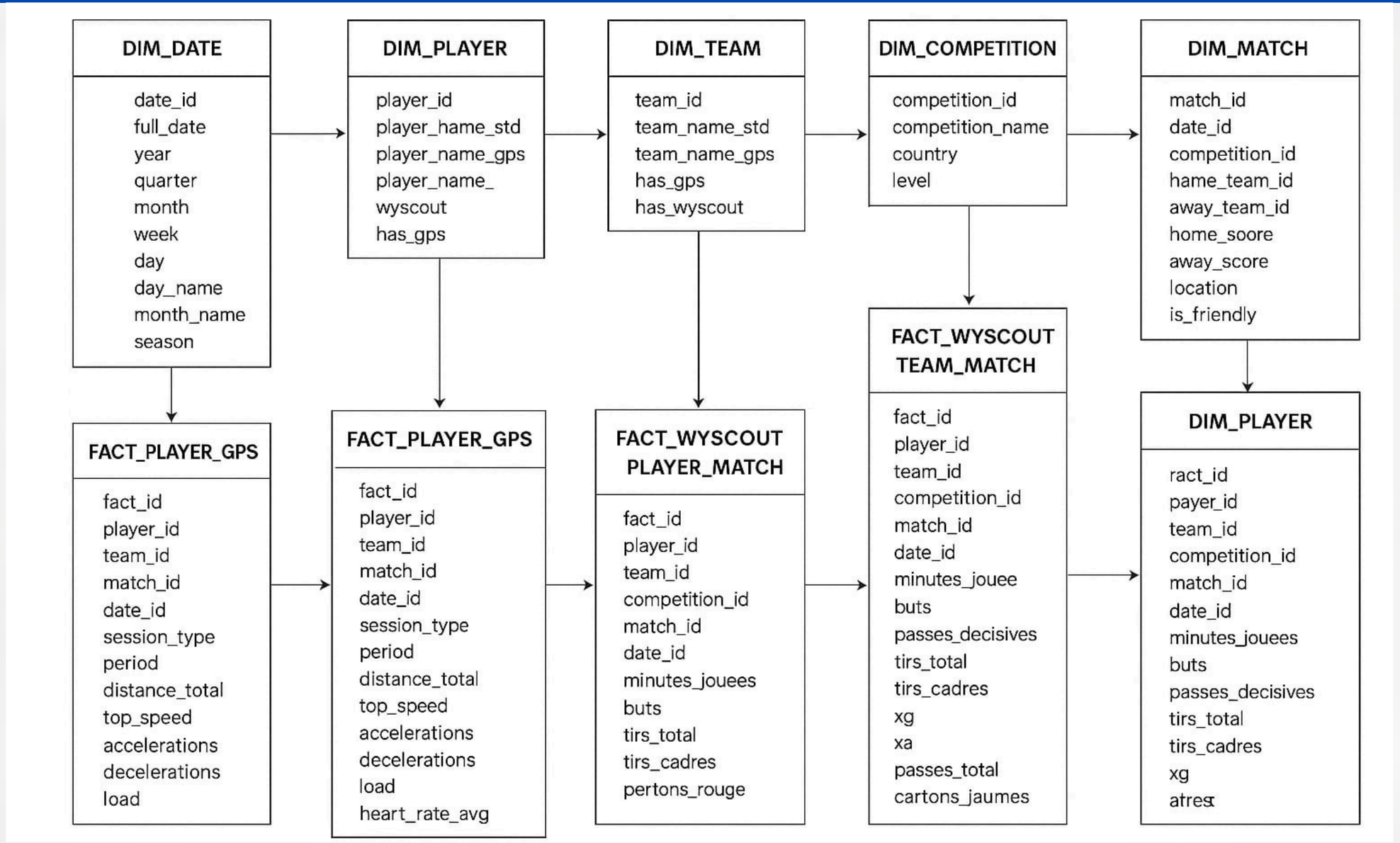
~~KAN-24~~

Keycloak authentication fully functional

~~KAN-25~~

Quickstart

DATA WAREHOUSE DESIGN SCHEMA



NAME	DESCRIPTION	ROWS (ESTIMATED)	SIZE (ESTIMATED)	REALTIME ENABLED	
dim_competition	No description	8	24 kB	✗	4 columns
dim_date	No description	3723	656 kB	✗	10 columns
dim_match	No description	2089	475 kB	✗	9 columns
dim_player	No description	147	40 kB	✗	6 columns
dim_team	No description	21	8 kB	✗	6 columns
exception_log	No description	18	7 kB	✗	7 columns
fact_player_gps	No description	10 011	968 kB	✗	16 columns
fact_wyscout_player_match	No description			✗	34 columns
fact_wyscout_team_match	No description			✗	23 columns

Problems Output Debug Console **Terminal** Ports

```
    [2266] 2025-10-01 | al nassar N/A al zawra'a | Asia. AFC Champions League Two
    [6] 2025-09-30 | al ittihad N/A al ittihad | N/A

  Statistiques:
    Dates différentes: 165
    Équipes (domicile): 14
    Équipes (extérieur): 110
    Compétitions: 8

======================================================================
  ✅ ETL DIM_MATCH TERMINÉ
======================================================================
```

BULLET POINTS

DATA QUALITY:

- Cleaned and validated data from ETL pipelines
- Consistent naming conventions and formats
- Referential integrity between fact and dimension tables

LOGICAL RELATIONSHIPS:

- Fact table linked to all dimension tables via foreign keys
- Supports complex queries like “Player performance vs. training load”
- Star schema ensures clear hierarchy for analytical processing

ANALYTICAL EFFICIENCY:

- Pre-aggregated metrics for faster query performance
- Optimized joins between fact and dimensions
- Supports API dashboards, comparisons, and trend analysis

WEB APPLICATION

Sports Analytics

- Dashboard
- Players
- Matches
- Training
- Reports
- Settings

Dashboard

User: Admin

Player Performance

Avg Goals: 2.3

Assists: 1.8

Match Stats

Total Matches: 87

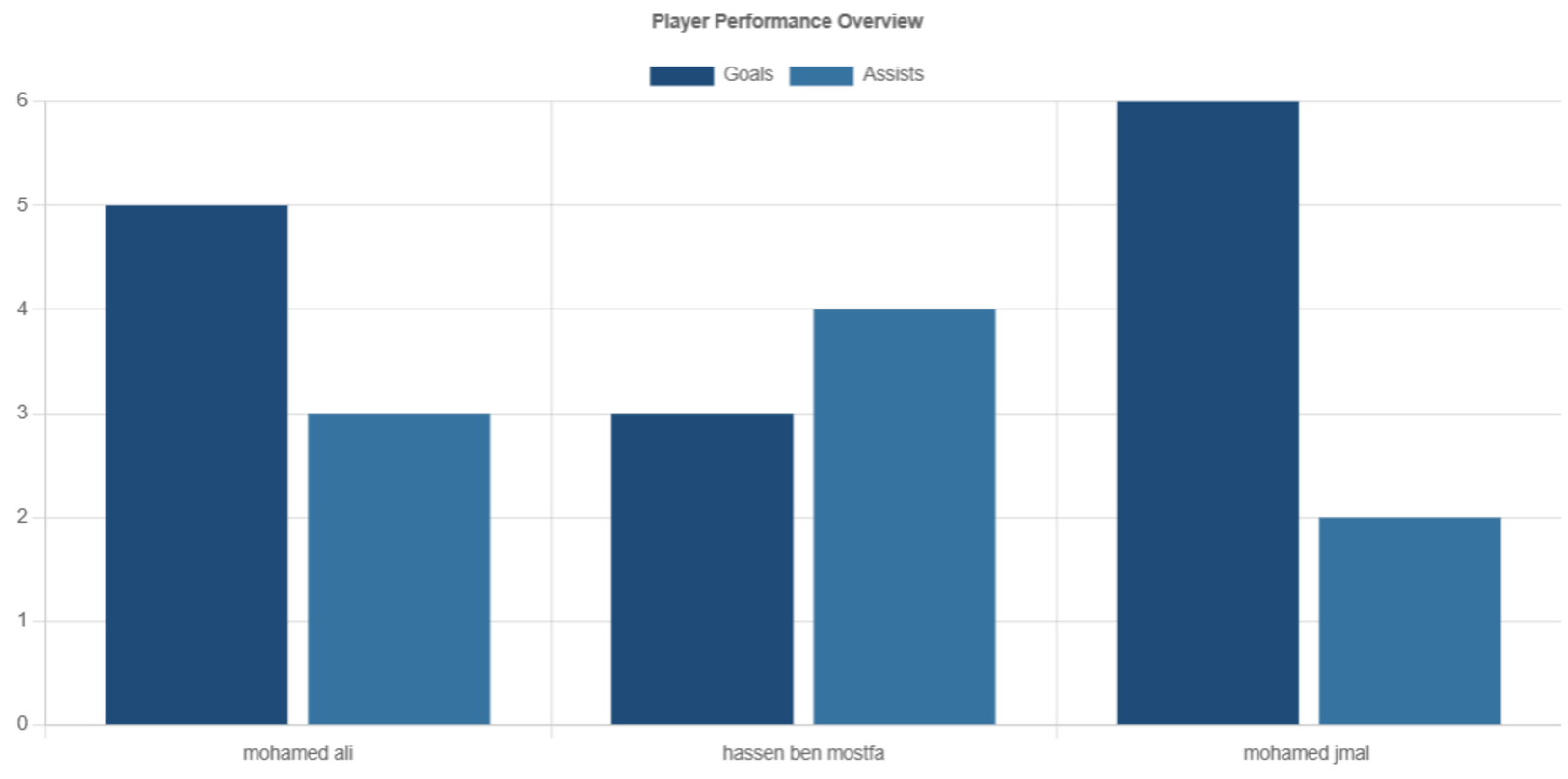
Win Rate: 73%

Training Hours

Avg per Player: 12h/week

Accuracy

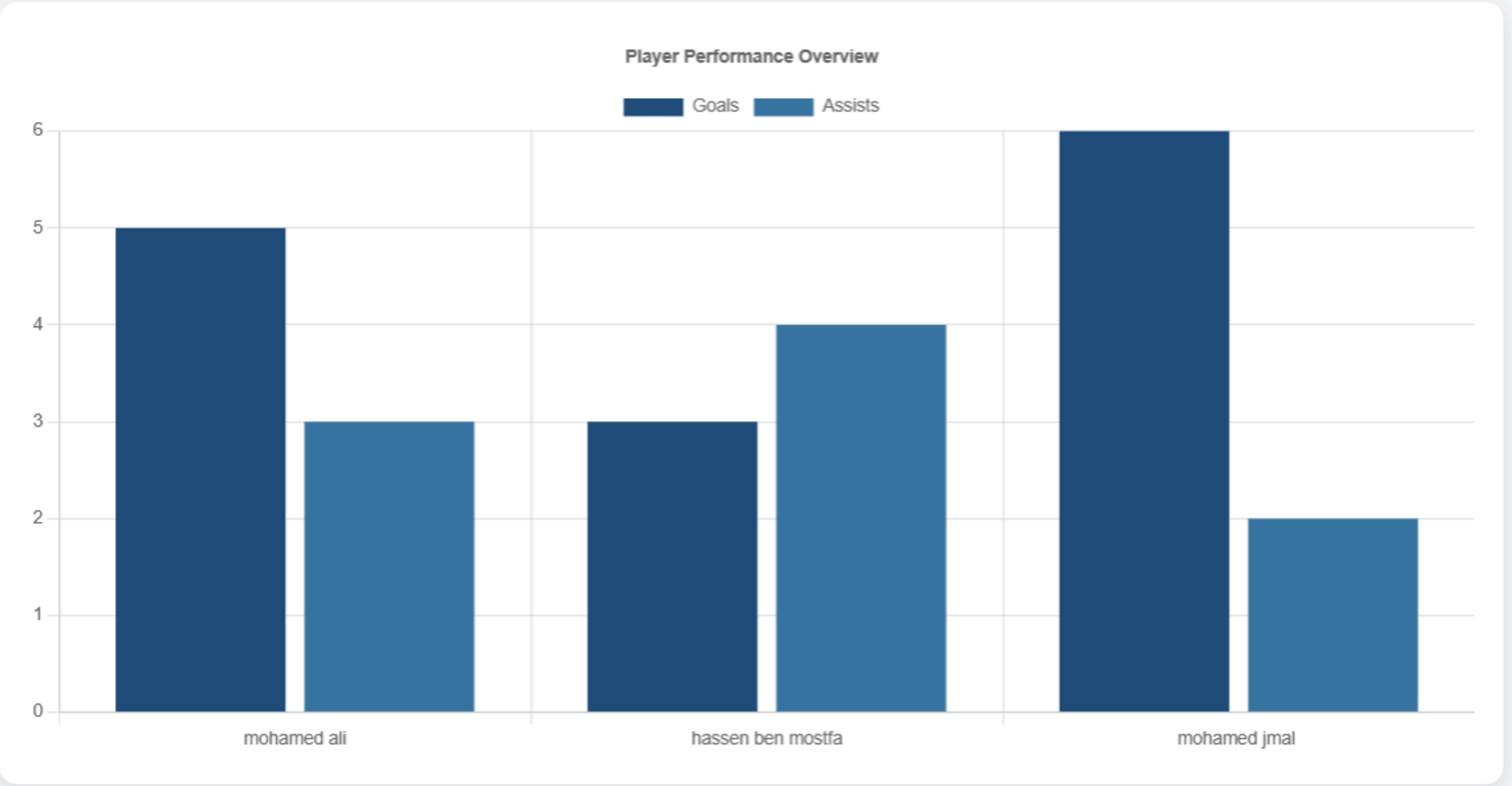
Data Validation: 91%



WEB APPLICATION

Sports Analytics

- Dashboard
- Players
- Matches
- Training
- Reports
- Settings



Player Name	Goals	Assists	Distance Run (km)	Calories Burned
mohamed ali	26	5	11.2	850
hassen ben mostfa	19	4	10.5	780
mohamed jmal	13	2	12.3	900

127.0.0.1:5500/test.html#

AI MODEL SELECTION & RATIONALE

1

LOGISTIC REGRESSION

- Predicts match outcome probabilities
- Simple, interpretable, baseline model for structured football data

2

RANDOM FOREST

- Handles non-linear interactions in player and team stats
- Robust to noisy and complex datasets

3

GRADIENT BOOSTED TREES (GBT)

- Best for tabular, structured data
- Improves accuracy and performance on complex patterns
- Efficient: faster to train than deep learning on medium datasets

AI Prediction Model

Training • ONNX Export • Web Worker Integration

1 Model Training

The AI model is trained using datasets containing player statistics, match history, and physiological data to **predict outcomes, fatigue levels, or in-game performance trends.**

Training Accuracy: **90%**

2 ONNX Model Export

Once trained, the model is exported to the **ONNX format**, allowing cross-platform use and optimization for inference across web and mobile environments.

File size: **18.4 MB**

Export Successful

3 Frontend Integration

The exported model is integrated into the frontend using **ONNX WebRuntime**, running inside a **Web Worker** for smooth, asynchronous predictions without UI lag.

Latency: **45 ms**

Parallel Workers: **3**

AI Model Pipeline

Real-time Prediction Example

Predicted Player Fatigue: Low (0.21)

Inference completed in **47 ms**

LESSONS LEARNED & INSIGHTS

Learnings
- End-to-end data flow
- Star schema efficiency
- AI inference in browser

Challenges
- Data volume
- ONNX export
- Frontend performance

Insights
- Integration
- Collaboration
- Process discipline

THANK YOU