

Premier League 2025/6 Dashboard

A learning project in Dash

August 2025

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Name | Signature | Position | Date |
| Prepared by | Richard Wood |  | Head Experimentation | 06 Aug 2025 |

## Executive Summary

This document outlines the design for a web-based dashboard for the Premier League's 2025/26 season. The dashboard, built with Dash (Python), will allow users to select home and away teams, view match details, and add or save commentary for specific games. The target audience includes analysts, football enthusiasts, and journalists who need to view and annotate match data. This project is a learning exercise in building a functional, responsive dashboard.

## Functional Requirements

Users must be able to select a ‘HomeTeam’ and an ‘AwayTeam’ from two separate dropdown menus, each containing all 20 Premier League teams.

The application will validate that the selected Home and Away teams are not the same.

The dashboard will automatically display the following metadata for the selected match:

* Match Date and Time
* The previous and next game for both the Home and Away teams.

An editable text area will be available for users to input commentary.

A savefunction will be included to persistently store user commentary in a database.

## Non-Functional Requirements

The application must be responsive and load within 1-2 seconds for typical user interactions.

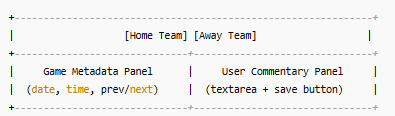
User commentary must persist across different user sessions.

The user interface should be clean and accessible.

## User Interface Design (Layout)

The dashboard layout is divided into three key sections to provide a clear and intuitive user experience:

* **Top Bar:**
* **Left Panel**: Game Info (loaded from CSV)
* **Right Panel**: TextArea for commentary + Save button



## Key Design Choices

### Frontend (Dash/HTML Layout)

* Use **Dash Core Components**:
  + dcc.Dropdown for team selection
  + dcc.Textarea or dcc.Input for user commentary
  + html.Button for Save
  + html.Div for dynamic content
* Use **Dash Callbacks** to:
  + Update match info when teams are selected
  + Enable/disable Save button as needed
  + Handle updates to user commentary

### Data Source

* Initial CSV file:  
  C:\Users\RichardWood\Documents\premiership\_football\data\epl-2025-GMTStandardTime.csv
* Assumptions:
  + The file contains 380 rows (each match).
  + Columns: Date, Time, Home Team, Away Team, etc.
  + Should be read at startup and stored as a pandas DataFrame.

### Persistence Strategy

Use a lightweight database (e.g., SQLite) with schema:

A computer code with text

AI-generated content may be incorrect.

Future-proof for multi-user or cloud persistence with PostgreSQL or Firebase.

### Navigation Logic

Identify match using (Home Team, Away Team) combination.

For “game before” and “game after,” sort by game date and look up adjacent games for each team.

## Architecture Diagram (Logical View)

A screenshot of a computer code

AI-generated content may be incorrect.

## Data Flow

User selects Home and Away Team  
→ Callback triggers.  
→ Match info is retrieved from in-memory CSV.  
→ Display match details + load any saved commentary from the DB.

User edits commentary and clicks Save  
→ Save commentary with ‘Match Number’ as key into the DB.  
→ Show confirmation to user.

## Technology Stack

| **Component** | **Technology** |
| --- | --- |
| Dashboard | Dash (Python) |
| Data Processing | Pandas |
| Persistence | SQLite |
| Web Server | Flask (via Dash) |
| Deployment | Local |

## Future Enhancements

* Calendar view of fixtures (month/week) using dash\_daq or fullcalendar.io integration
* Commentary history (versioning)
* User login and personalization
* REST API for external access

## Risks and Mitigations

| **Risk** | **Mitigation** |
| --- | --- |
| Data file structure changes | Validate and preprocess input CSV |
| User commentary collision | Add timestamp or user-specific key in future |