How football player transfers change the win or loose landscape in europe.

DaTeam :
Georg SCHWABEDAL 328434
Mathurin CHRITIN
Amene GAFSI

30 May 2025

Figure 1: EPFL logo

1 Introduction

Football transcends sport, uniting fans, analysts, and professionals through its blend of competition and strategy. Our project transforms this passion into an interactive data visualization platform, exploring the intricate relationships between match performance, player transfers, and financial strategies in European football from 2008 to 2016. By weaving together data-driven insights and engaging visuals, we aim to illuminate how clubs evolve through competition and investment.

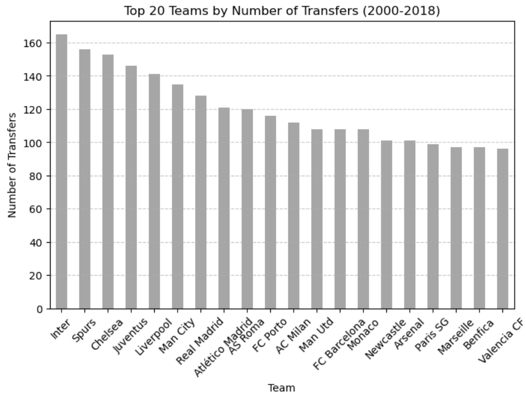


Figure 2: Top teams by transfer activity

The heart of our platform is a map-based interface that allows users to navigate club histories through match outcomes (wins, losses, draws) and transfer networks. We integrate two comprehensive datasets: one detailing match statistics from 2008 to 2016 and another covering player transfers from 2000 to 2018. Clubs are geolocated on a zoomable European map, with transfer flows visualized as connecting lines and team-specific details displayed in dynamic side panels. This creates a cohesive narrative of football's complex ecosystem.

Our platform targets diverse audiences: football fans eager to explore team trends, data analysts seeking strategic patterns, and scouts or managers evaluating transfer impacts. By balancing statistical depth with intuitive design, we aim to inform and spark meaningful discussions about football's modern evolution.

2 Work Structure

Guided by course milestones, our development process ensured a steady and collaborative workflow. As football enthusiasts from different countries, we chose to focus on a map-based visualization to track player transfers and their impact on team performance across different regions. We began by identifying robust datasets, performing initial preprocessing, and sketching a prototype that included a map, global statistics, a side panel for team details, and visualizations of transfer flows.

Our technical journey started with a minimum viable product (MVP) website, featuring a navigation bar, page placeholders, and hyperlinks. This early version allowed us to refine critical design choices, such as color schemes, single-page versus multi-page layouts, and feature scope. A dedicated team meeting helped resolve these questions, enabling us to modularize tasks based on complexity, time requirements, and course progress. Work was distributed among team members for concurrent development, followed by a thorough collective review to ensure consistency and accuracy across code and reporting.

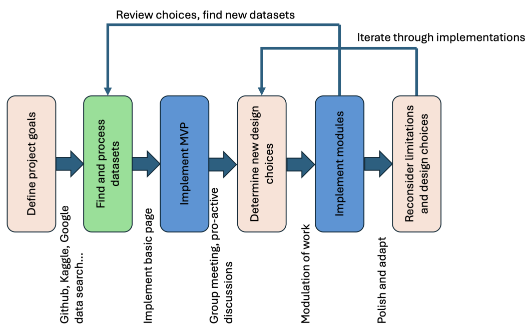


Figure 3: Project workflow overview

3 Technical Challenges and Implementations

To realize our vision, we divided the project into modular components, each assigned to a team member. Below, we detail these modules, their technical implementations, challenges, and limitations, providing deeper insight into our approach.

3.1 Data Loading and Processing Pipeline

Implemented by: Amene & Mathurin.

Technical Overview: Our data pipeline transforms a 29,000-match SQLite database and thousands of transfer records into optimized JSON and CSV files for fast, browser-friendly rendering. It handles extraction, cleaning, merging, and aggregation to support an interactive, data-heavy front end.
Challenges: We faced naming inconsistencies (e.g., "Man Utd" vs "Manchester United"), large data volumes unsuitable for client-side processing, and the need to compute metrics like wins or spending offline. Ensuring accuracy while filtering and merging was critical.
Solutions: Using Python and SQL in Jupyter, we standardized names with mapping tables, filtered for top-tier clubs (2008–2016), and precomputed key statistics. Additional info—logos, coordinates, was linked to ensure consistency. This setup offloads heavy computation to preprocessing, ensuring smooth web interactivity.
Limitations: The pipeline outputs are static, requiring reprocessing for updates beyond 2016. Manual corrections limit scalability, and some smaller clubs may be excluded. Still, the result is a performant and reliable data layer for our visualizations.

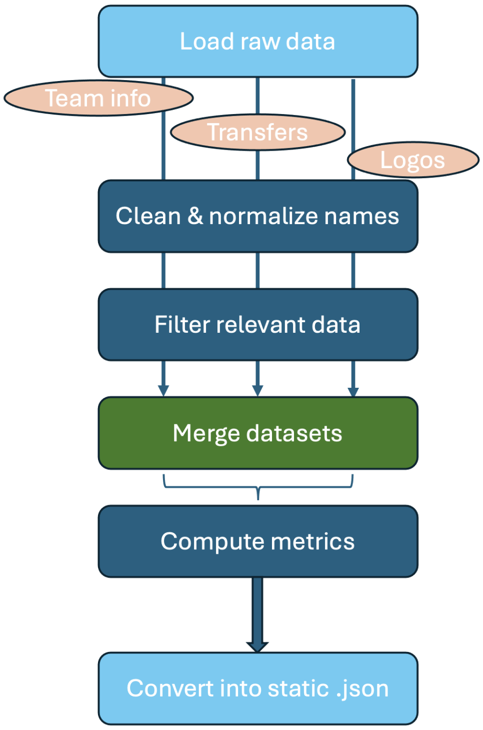


Figure 4: Data-processing steps

3.2 Team Logo Integration

Implemented by: Georg

Figure 5: Major teams logos

Technical Overview: Club logos serve as key visual anchors across the interface, used as custom markers on the map, in the team selection scrollbar, and within the side panel. Their integration improves visual clarity and user engagement by offering instant recognition and a cohesive design language throughout the platform.
Challenges: Significant inconsistencies in club naming across datasets (e.g., "Bayern Munich" vs. "FC Bayern München") made direct logo assignment unreliable. Our external logo source, a GitHub repository, included logos for top-division clubs from 11 countries but was incomplete for lesser-known teams and those with historical name changes. Additional challenges included broken image paths, flickering UI elements during load, and duplicated or outdated logo files.
Solutions: We created a Python mapping dictionary to standardize team names across datasets, exported it as a CSV, and converted it to JSON for browser use. To simplify linking logos, we matched their file names to team folder names. We manually fixed missing or incorrect logos by renaming files, adding aliases, and using fallback images.

To avoid flickering or broken images, we optimized file paths and added basic preloading. Logos were used as custom Leaflet markers and also displayed in the sidebar and scrollbar with HTML image tags and tooltips for clarity.

Limitations: Some minor clubs still rely on placeholders due to unavailable logos. The manual nature of the name-to-logo mapping process is not easily scalable for larger or updated datasets. Since the logos are static assets, updating them for future seasons or branding changes requires manual replacement and testing.

3.3 Introduction Page

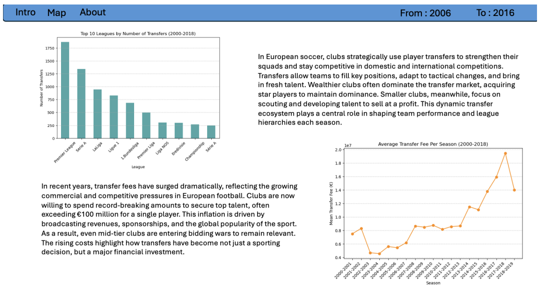
Implemented by: Amene and Mathurin.

Technical Overview: The introduction page serves as the landing interface of the platform. It presents a high-level summary of the project and guides users toward exploring its features. The page highlights the purpose of the visualizations and acts as the first point of contact for users arriving at the site.

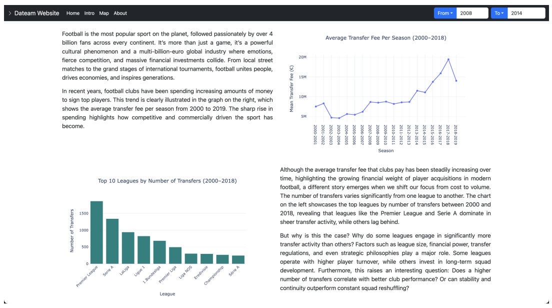
Challenges: The main challenge was to balance clarity and engagement, ensuring the introduction was informative without being overwhelming. Layout responsiveness across devices and clear visual hierarchy were essential to maintain usability.

Solutions: We designed a clean, minimal interface with concise text and visual cues to direct users. The page introduces the project goals and prompts users to explore the map or select a team. Basic responsive techniques were used to ensure readability and layout stability on different screen sizes.

Limitations: The current introduction lacks richer media elements, such as animations or onboarding tutorials, that could enhance engagement. Future versions could integrate short videos or walkthroughs to improve the user onboarding experience.



(a) Initial introduction sketch



(b) Final introduction

Figure 6: Comparison between initial introductions and final result

3.4 Interactive Map

Implemented by: Georg

Technical Overview: The interactive map, built with Leaflet.js, is the platform's centerpiece. It displays club locations as custom logo markers on a zoomable European map. Users can explore teams geographically, visualize transfer connections, and access detailed team data through interactive clicks. This spatial interface enhances contextual understanding and overall engagement.

Challenges: Team stadium coordinates were not incldued with the logos. Visual clutter arose in dense regions (e.g., London, Milan) due to overlapping markers. Integrating dynamic overlays such as transfer lines with Leaflet's tile-based rendering demanded careful performance management.

Solutions: We compiled accurate club coordinates from external references and standardized team names to ensure consistency across components. Marker clustering and slight coordinate offsets reduced visual overlap, with manual adjustments in crowded cities. Using Leaflet's API, we implemented custom logo markers, click event handlers, and polylines for transfers. In some cases, we used D3.js to draw SVG paths for better visual control, ensuring alignment during zoom and pan interactions.

Limitations: At wide zoom levels, the map can appear cluttered since all clubs are shown by default. Planned features like league-based filtering were not implemented.

(a) Initial Map sketch

(b) Final product

Figure 7: Comparison between initial map and final product

3.5 Side Panel Interface

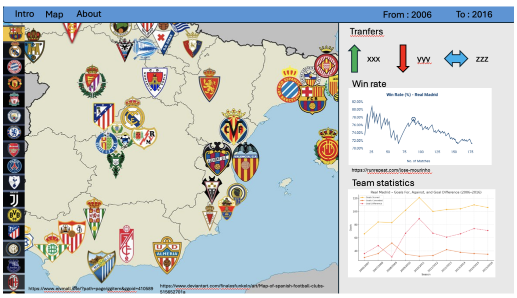
Implemented by: Georg & Amene.

Technical Overview: The side panel acts as a dynamic dashboard for team-specific data, displaying the selected club's name, logo, win/draw/loss charts, and transfer statistics. This focused view allows users to explore one team in detail, enhancing narrative depth and interactivity.

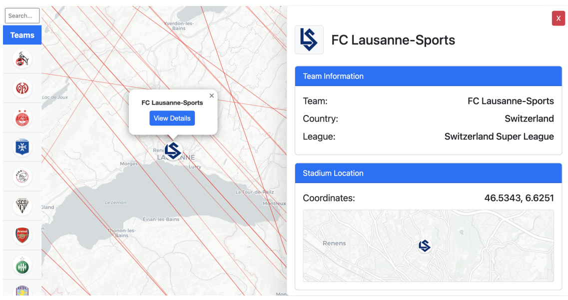
Challenges: Synchronizing the panel with selections from both the map and the team list required robust state tracking. Accommodating diverse content: logos, text, and SVG charts, within a limited UI space without visual clutter proved difficult. Early issues with data mismatches led to incorrect or missing content, and loading rich media like charts and logos created occasional performance bottlenecks.

Solutions: We implemented the side panel as a toggleable HTML/CSS element updated via JavaScript event handlers. Using standardized team IDs, we ensured consistent data retrieval across components. The panel layout was split into logical subsections: overview (logo and name), performance charts (via D3.js), and transfer stats. Charts updated smoothly using D3's data binding, and optimized image loading prevented lag. CSS scrollability enabled handling of larger content blocks, while caching reduced unnecessary reloads.

Limitations: Advanced features like detailed transfer breakdowns or team comparisons were not fully implemented. The panel layout is static and scroll-dependent. This limits usability for comparing multiple clubs and adds friction for data-heavy teams.



(a) Initial side pannel



(b) Final side pannel

Figure 8: Comparison between initial side pannel and final side pannel

3.6 Team List Panel

Implemented by: Amene & Georg

Technical Overview: The team list panel on the left side of the map displays a complete scrollable list of all teams in the dataset, providing users with a reliable alternative to selecting clubs directly from the map. While the map offers geographical context, accurately locating a specific team can be difficult due to overlapping markers or dense regions. The team list ensures that every team is easily accessible regardless of map visibility.

Challenges: Locating teams purely through the map interface proved unintuitive in many cases, particularly for regions with high team density or for users unfamiliar with club locations.

Solutions: To address these usability issues, we introduced a powerful search bar integrated into the top of the team list panel. This became a central navigation tool, allowing users to instantly filter the list by typing the name of a club. The search functionality drastically improves accessibility, especially for users who know which team they want to explore but struggle to find it spatially on the map. The list items are connected to their corresponding team data using standardized IDs, enabling accurate synchronization across the interface. The full list remains scrollable and alphabetically ordered for manual browsing.

Limitations: A primary limitation of the search bar is its reliance on exact string matching. Due to variations in team naming conventions, such as "Manchester United" vs. "Man Utd", users may encounter difficulties unless they use the precise format stored in the dataset. This can reduce the effectiveness of the search for casual users. A possible future enhancement is to integrate fuzzy search to extend the current support for partial matches and accepted name variants for each team.

(a) Initial team list pannel sketch

(b) Final team list pannel

Figure 9: Comparison between initial and final team list panel

3.7 Team Performance Visualizations

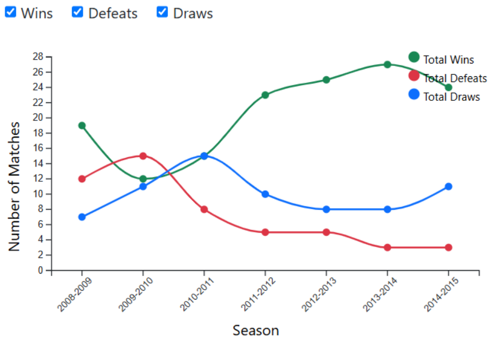
Implemented by: Amene

Technical Overview: Built with D3.js, these charts show season-by-season outcomes: wins, draws, and losses from 2008 to 2016. They offer users a clear view of each club's on-field performance and its evolution over time. These charts are dynamic to the years selected on the navigation bar: if a user is interested in a specific time frame, selecting the desired year interval will automatically update the visualization to reflect only those seasons. Additionally, the charts are interactive with respect to the metrics displayed: users can selectively choose which performance indicators to view. For example, if a user is only interested in the number of wins per season and the total amount of money spent, they can simply tick those metrics to tailor the visualization accordingly.

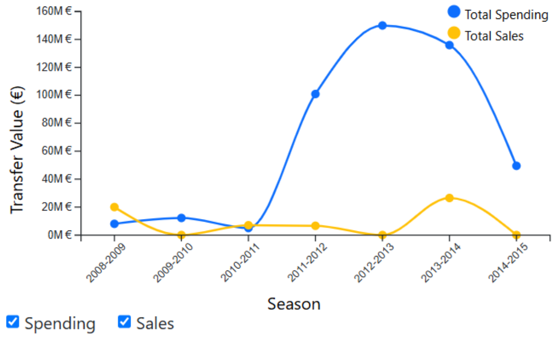
Challenges: Summarizing over 25,000 matches efficiently was essential to avoid browser lag. Adapting charts to different team scales and fitting them into the side panel layout required careful design and processing.

Solutions: We precomputed per-season stats and saved them as compact JSON files. D3.js rendered charts with adaptive scales, clear color coding (e.g., green for wins), and tooltips. Data binding enabled smooth updates when switching teams, and the charts were embedded in the side panel for consistency.

Limitations: The charts focus on basic metrics and a fixed period (2008–2016). Advanced analytics like goal difference or direct comparisons between teams are not included, limiting broader analysis.



(a) PSG 2008-2014 Win/Loss/Draw values



(b) PSG 2008-2014 Transfer values

Figure 10: FC Köln description metrics in side pannel - Team Performance

3.8 Transfer Visualizations

Implemented by: Mathurin

Technical Overview: Transfer visualizations show player movements as lines connecting clubs on the map, illustrating a selected team's network of incoming and outgoing transfers. This feature highlights how players and financial resources move across Europe, revealing strategic patterns in club behavior.

Challenges: The dataset includes thousands of transfers over nearly two decades, which risked overloading the interface. Filtering for only relevant transfers was crucial. Drawing multiple lines between clubs—especially in dense regions—led to overlap, and showing directionality (seller vs. buyer) required thoughtful visual encoding. Rendering too many polylines or paths at once also posed performance concerns.

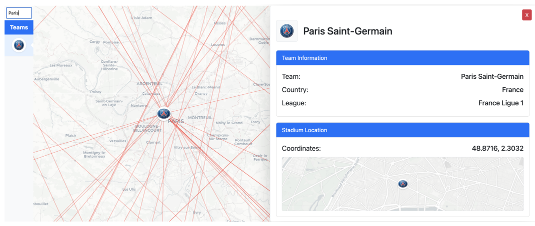


Figure 11: PSG 2008-2014 transfers

Solutions: We aligned transfers to standardized team IDs and filtered results dynamically based on user selection. Transfers were visualized using Leaflet polylines or D3 SVG paths, with curves to reduce overlap and color coding to distinguish incoming from outgoing flows. Line thickness or opacity represented transfer volume. Interaction handlers allowed real-time updates when teams were switched, and hover effects revealed additional info such as number of players or transfer fees.

Limitations: A global transfer network view was excluded to avoid clutter. Transfer size is only implied (via visual encoding), and directionality cues like arrowheads are subtle. Advanced filters—such as by year, position, or fee—were not fully implemented, limiting deeper exploration.

3.9 Use of D3.js and Leaflet.js

Implemented by: Mathurin & Amene & Georg

Technical Overview: The project combines Leaflet.js for geospatial rendering and D3.js for data visualizations to deliver an engaging user experience. Leaflet manages map interactions and geographic markers, while D3 handles charts and optional SVG overlays, leveraging each library's strengths.

Challenges: Aligning D3's SVG elements with Leaflet's tile-based map during zoom and pan was technically demanding. D3's DOM-based rendering can also impact performance with large datasets, requiring careful division of responsibilities. Maintaining consistent styling between charts and map elements introduced CSS and color coordination challenges.

Solutions: We used Leaflet for interactive markers and transfer lines, and D3 for performance charts and dynamic overlays. D3's output was synchronized with Leaflet using projection plugins or transforms. Scale functions and data binding enabled smooth transitions in charts, while Leaflet's native API ensured fast, fluid map interaction. Shared color palettes and styling rules maintained visual consistency across components.

Limitations: Advanced interactions, such as brushing charts to highlight corresponding map elements, were not implemented due to integration complexity. Some event handling between the two libraries overlaps and could be refactored. Though the system is optimized for typical use, heavy rendering (e.g., all transfers at once) could strain D3. Despite these limits, the libraries were successfully combined to support a responsive and modular visualization.

4 Conclusion

This project transformed complex European football data into an interactive and engaging platform. Users can explore club performance, transfer activity, and strategic patterns through a map-based interface, dynamic charts, and a responsive side panel. While some features, such as multi-team comparisons and advanced filters, remain unimplemented, the current version offers a technically sound and visually compelling experience.

Throughout the project, we navigated challenges in data integration, library coordination, and frontend performance. These experiences strengthened our skills in data processing, web-based visualization, and collaborative development. Most importantly, the project highlighted the value of visual storytelling in making intricate systems accessible and intuitive.

We would like to thank Professor Laurent Vuillon and the entire course staff for their guidance. The structure and clarity provided by the milestone framework were instrumental in shaping our progress and helping us maintain a focused scope.

5 References

- Mathien, H. (2016). *European Soccer Database*. Kaggle.
  https://www.kaggle.com/datasets/hugomathien/soccer
- Slehkyi, S. (2018). *Football Transfers 2000–2018*. Kaggle.
  https://www.kaggle.com/code/slehkyi/football-transfers-2000-2018
- Hopman, L. (2024). *Football Team Logos 2024/2025*. GitHub.
  https://github.com/luukhopman/football-logos
- Creamer, C. (n.d.). *European Football Logos by League*. SportsLogos.net.
  https://www.sportslogos.net/leagues/list_by_category/31/European-Football-Logos/
- Antimoni, A. (2019). *Football Stadiums Dataset*. Kaggle.
  https://www.kaggle.com/datasets/antimoni/football-stadiums
- Jokecamp. (2017). *Stadiums with GPS Coordinates*. GitHub.
  https://github.com/jokecamp/FootballData/blob/master/other/stadiums-with-GPS-coordinates.csv
- Wikipedia Contributors. (2024). *List of top-division football clubs in UEFA countries*. Wikipedia.
  https://en.wikipedia.org/wiki/List_of_top-division_football_clubs_in_UEFA_countries

- Hopman, L. (2024). *Issue Tracker for Football Logos Repository.* GitHub. https://github.com/luukhopman/football-logos/issues