



Design and Development of an Interactive Football
Statistics Web Application with a Focus on Data
Visualisation

By

Oliver O’Keeffe

N00221912

DL836 – BSc(Hons) Creative Computing

Supervisor: Cyril Connolly

Second Reader: John Dempsey

Declaration of Originality

I declare that this project is my own work and that it has not been submitted for any other assessment. Where the work of others has been used, it has been fully referenced and acknowledged.

I confirm that I have adhered to the academic integrity guidelines of my institution.

Student Name: Oliver O’Keeffe

Student ID: N00221912

Module: DL836 – BSc(Hons) Creative Computing

Signature: _____ Oliver O’Keeffe

Date: _____ 30/04/2026

Table of Contents

1	Introduction and Project Context	1
1.1	Project Aim and Objectives	1
1.1.1	Objectives	1
1.2	Success Criteria and Scope	1
2	Research and Background	2
2.1	Introduction	2
2.2	Data Visualisation in Sports	2
2.2.1	Definition and Purpose of Data Visualisation	2
2.2.2	Principles of Effective Data Visualisation	3
2.2.3	Historical Development of Data Visualisation	3
2.2.4	Sports Data Visualisation	4
2.2.5	Types of Sports Data (Spatiotemporal and Statistical)	4
2.2.6	Visualisation Techniques Used in Sports	5
2.3	Data Visualisation in Football	5
2.3.1	Challenges and Limitations in Sports Data Visualisation	6
2.4	Machine Learning in Sports Analytics	6
2.4.1	Overview of Machine Learning	6
2.4.2	Types of Machine Learning Algorithms	7
2.4.3	Real-World Applications of Machine Learning	7
2.4.4	Machine Learning in Sports	7
2.4.5	Challenges and Future Directions of Machine Learning in Sports	8
2.5	Technologies and Environments	8
2.5.1	Overview of Data Visualisation Technologies	8
2.5.2	Python as a Data Analysis and Machine Learning Tool	8
2.5.3	Dash and Plotly for Interactive Visualisation	8
2.5.4	Integrating Machine Learning with Visual Analytics	9
2.5.5	Benefits of Interactive Dashboards in Sports Analytics	9
3	Requirements Analysis	9
3.1	Requirements Gathering & Modelling	9
3.2	Functional Requirements	10
3.3	Non-Functional Requirements	10
3.4	User Personas	11
3.5	User Cases	11
4	System Design	11
4.1	System Architecture	11

4.2	Interface and User Experience Design	12
4.3	Data Visualisation Design	14
4.4	Database Design	16
5	Implementation	17
5.1	Development Methodology	17
5.2	Frontend and Backend Development	17
5.3	Data Integration and API's.....	19
5.4	Challenges and solutions.....	22
6	Testing and Evaluation	23
6.1	Test Plan	23
6.2	Testing Results	24
6.3	Usability Testing	24
6.4	System Evaluation	25
7	Project Management	26
7.1	Project Planning and Organisation	26
7.2	Tools and Workflow	26
8	Conclusion And Future Work	27
8.1	Summary of Findings	27
8.2	Limitations	27
8.3	Future Improvements	27
9	References	29
10	Appendices	30

1 Introduction and Project Context

1.1 Project Aim and Objectives

The aim of this project is to create a website that can be used by all that are either involved, or interested in football. They will be able to look at in-depth interactive charts and see AI predictions for upcoming games and stats. All achieved by using an API to retrieve the data.

1.1.1 Objectives

- To integrate a football statistics API to retrieve real time data that include results, fixtures, league tables and player stats
- Develop and design a user-friendly website for navigating all stats
- Create interactive charts that to visualize key stats and give more depth to them
- Implement filter and search features to give the user a more efficient experience
- Develop a predictive AI feature that gives predictions for matches, league standings and top scorers/ assisters
- Test the website for data accuracy, usability and performance

1.2 Success Criteria and Scope

This project will be deemed successful if the website has met the goals that I have set. This includes both functionality and a good user experience. It must successfully retrieve and display the correct data from the external API that I have chosen to use, including real time data - such as fixtures, results, player stats and league standings.

The website should show this data through clear and interactive graphs and charts. These will allow the user to easily read the data and understand the trends and comparisons. Interactivity should include filtering, searching and hover features.

The predictive feature will be seen as a success if It can more often than not predict the correct winner of a match. This will be based off of historical and present day data gathered through the API.

The scope of this project is limited to the development of a web-based football statistics application focused on data visualisation that I will be making. The system will allow the user to explore real time and historical data to do with league, team, and player data but will not include advanced features such as user account systems or highly complex machine learning prediction models unless there is extra time to try and implement it. The project will be completed within the timeframe and with the available resources.

2 Research and Background

2.1 Introduction

Over recent years, data analytics has become more prominent in professional sports. Data in sport originates from different people and areas. There are, for example, the match statistics themselves, sensors that the players wear, video analysis and tracking systems. The data that is received gets used in different ways, depending on who is looking at it. For example, players will look at data to see how they can improve in certain areas, while teams can use data to see what type of player they need to sign for their system.

This research section will look into data analytics as it has become an integral part of all sport, but specifically football. Data analytics is being used, not only for the more obvious benefits such as tactical analysis and performance optimisation, but also for injury prevention – this has become especially important as clubs spend large amounts of money on new players whom they must keep match fit for as long as is possible.

Specifically, data visualisation is key, as reading raw data is difficult to understand and interpret. Therefore, making data visual easier to understand is paramount for understanding. It also makes it easier to see things such as trends and patterns in certain types of data and saves time by it being easier to read. Issues can arise though if data is visualised poorly, which can further lead to misunderstandings.

This section will examine data visualisation with machine learning and how it is being used in sports - specifically football. It will go into three main concepts. (1) Data visualisation (2) Machine learning algorithms and (3) The technologies and environments that are used for machine learning and data visualisation in sport.

2.2 Data Visualisation in Sports

2.2.1 Definition and Purpose of Data Visualisation

Data visualisation is the graphical representation of data, which can be shown through plots, infographics, charts, and animations (IBM Think, 2024). Raw data needs to be converted into visualisations because data in its original form can be hard to read and understand – furthermore, large volumes of numbers often don't make sense at first glance and make it difficult to spot patterns or understand the context (RudderStack, 2025). Visualising raw data allows the reader to spot underlying patterns and gain context, making information more interpretable. This ultimately assists researchers, readers, and sports analysts to understand and use the data more effectively.

2.2.2 Principles of Effective Data Visualisation

To have effective visuals in data visualisation, there are certain principles that are followed. These principles are used to maintain clarity and accuracy within the visuals. One of the main and most basic principle is called Diagram First, which is to prioritise what information you want to be visualised. This is as fundamental as thinking if you want the graphic to be a comparison of data, or ranking of a certain piece of data. Colour Encoding is one of the principles that is most important, as a graphic colour always means and represents something. The use of colour is especially powerful in these graphs because it is an easy and digitally cheap way to separate data and make it easy to understand and compare.

Simple visuals with detailed captions, is a principle that can easily be overlooked. One must never overcomplicate the graph, remembering that most graphs are only displayed for a few metrics at most. This is done so the graphs are not overcomplicated, which would then be counterintuitive. The whole purpose of these graphs is, of course, to make data easy to understand.

Getting a second or even third opinion is imperative, as this gives you insight into what the readers may really think of your graphics. It will help you see things that you wouldn't have before, as you would inevitably examine the graphs differently to someone who may not be as focused on, or familiar with, data visualization.

Following these principles is integral to creating good graphs (Cairo *et al.*, 2020).

2.2.3 Historical Development of Data Visualisation

Early visual representation first dates back to the 2nd century in Egypt. They organised astronomical information to try and help with navigation. In the last 30 years, improvements really began apace. (During this time, it was mainly used to just record the data, not analyse it). Data visualization began to develop and take off in the 17th century, with Rene Descartes introduced coordinate systems. Originally this was just a way for people to represent *quantitative* data, but later on was confirmed to be a valid way to represent data.

It wasn't then until the 18th and 19th century that graphs such as pie and bar charts came into being. These graphs were either invented or improved by a Scottish social scientist, William Playfair. These charts and techniques didn't become recognised academically though for another century by Iowa state university in 1913. The first time we saw proper graphics digitally was in 1984, when Apple introduced the first computer that was focused on graphical information. (Few, 2020) The first ever data analyst in football was Charles Reep in 1950 ("Data and decisions in soccer," 2022).

2.2.4 Sports Data Visualisation

Visualising data is an essential part of modern-day sports. Sports generate a lot of data, and most of this data is used and looked at in depth. It would be nearly impossible to work with all of this data if it was raw, and not converted in visual graphs. Every part of a sports team looks at data, whether that is the players, coaches or data analysts themselves. They are all looking at this data to try and spot trends and patterns all for different reasons. Players look at it to see what areas of their game they could improve; coaches look at it to see how to get the best out of all the different players. Data analysts look to ascertain which profile of players would best suit the squad.

A very common way the data is used and visualised is comparing teams or players. This is an excellent way to see which players in a certain position perform at certain things. In modern sports most decisions made by anyone involved with a team, are made by the backing of data. Therefore, ensuring data is easy to understand is vital for all sports. We will now look at the different types of data in sports, namely, statistical and spatiotemporal data (Du & Yuan, 2020).

2.2.5 Types of Sports Data (Spatiotemporal and Statistical)

Sports data is not tied to one type of data - there are many different forms in sports. The different types of data all have different characteristics and are used in different ways in sports. Two of the main types of data we see in sports are statistical and spatiotemporal.

Statistical data is numerical event-based data. This data is retrieved from specific events that occur in a match. This includes data such as goals, points, number of fouls and possession. This is important data, as it is easy to compare these numbers and then set players apart from each other and see what players are doing certain things for better or worse. This data tends to be more visually basic and easily understandable on graphs such as bar charts (Du & Yuan, 2020).

Spatiotemporal data is somewhat less discrete; it is the capture of positions and movements of the players over time during a match. Examples of this data are tracking the player's position, trajectory of the players and team shape and how that changes over the course of a game. The tracking data is usually gathered using special cameras and sensors that the players wear. This makes it more complex to gather and display as it is not a specific number, such as how many points a player achieved. It can be very valuable though – for example, a manager could look to see where the players are positioned on average. This allows him to tweak certain things about a formation (Du & Yuan, 2020).

2.2.6 Visualisation Techniques Used in Sports

There are also different techniques that are used to create graphs depending on what data you are using.

Good techniques that are commonly used for statistical data are, bar charts, radar plots, scatter plots and rankings. These are used for stats such as comparing top goalscorers or comparing players at specific things to see who performs best. These techniques are valuable as they are easy to understand and give clarity to the reader on what they are looking at. These techniques work well for statistical data as they are fixed numbers that make it easy to compare. That is what these charts are best for – comparison (Du & Yuan, 2020).

Spatiotemporal data requires different techniques though as the data that is being displayed is dissimilar. Some examples of these would be heat maps and positional maps. These plots capture movement, position and space. These can be used for readers to spots patterns and specific behaviours which makes them more complex to read and interpret (Du & Yuan, 2020).

Sports visualisation does have its challenges however. Due to there being so much information gathered, visual clutter and information overload can easily occur if not looked at carefully. Such mistakes can then lead to misunderstandings in the data if visuals are cluttered. Techniques aren't just specific types of graphs – rather a technique that is used in all graphs is interactivity. This includes filtering, sliders and zoom features. These allow readers to see what specific data that they need, if they only require to see data for certain players. These techniques are most commonly used in football (Perin et al., 2018).

2.3 Data Visualisation in Football

Football is a sport in particular which has numerous data batches that get retrieved after every match. Multiple different data is gleaned from the two types of data mentioned previously. Statistical data from the goals, assists, saves, touches, passes and many more, can all be looked at in individually and in depth. Spatiotemporal data on the other hand is from the players position, speed and team shape. These portions of data are gathered from football players all wearing sensors that allow teams to capture this information (“Data and decisions in soccer,” 2022).

Some common data visualizations in football include radar plots which are used when either examining stats for one player, or comparing players. It is an easy way to ascertain which player is performing better at certain things, by virtue of how the radar is laid out. Two similar graphs that are usually used simultaneously, are ‘passing networks’ and ‘average position’. These show how the team looks regarding its shape, throughout the match and with the passing networks graph, we can see all the players passing volume and to whom they are passing to. Shot maps and xG (expected goals) maps are very suitable for accessing players shooting. If a player has a high xG (expected goals) but not

the goals to back it up, you can then look at their shot map to see if they need to adjust where they are shooting (Du & Yuan, 2020).

These graphs and their information assist all club members. The coach can use them for tactical decisions and player performance reviews. It also helps them and the coaching staff to analyse the opposition before a game and to see where their weaknesses lie. The Scout can use them for player recruitment to see what players could be useful to sign. In truth, these graphs all help with recognising patterns and clarity for everyone involved (Perin *et al.*, 2018).

2.3.1 Challenges and Limitations in Sports Data Visualisation

Issues start to arise with sports data visualisation due to the large amount of data that is inevitably gathered. In sports, data that is gathered tends to be very dense and high-dimensional. Spatiotemporal data is specifically a very dense form of data and with this you then can get visualizations that are overwhelmed and hard to understand, as you get visual clutter and information overload (Perin *et al.*, 2018).

With information overload and visual clutter, you then additionally get poor design. This directly leads to reduction in clarity and so readers will get the incorrect information from the visuals. This is a problem in sport, as not only are most big decisions made once the data is backed up in modern sports, they are also time sensitive. Getting these decisions incorrect because of cluttered graphs can have serious consequences, as they can disturb the tactics and performance of the team (Perin *et al.*, 2018).

As mentioned previously, with the principles of data visualization, not everyone can understand visuals as well as the data analyst. This means that every graph has to be made as easy to read and to understand as possible, so all may benefit. Errors with the technology used also has to be considered, as if a sensor is faulty, then this can lead to incorrect data (Cairo *et al.*, 2020).

2.4 Machine Learning in Sports Analytics

2.4.1 Overview of Machine Learning

Machine learning is one of the many strands of AI. Its purpose is to learn patterns from training data. This will then allow computer systems to perform better on tasks without having to be programmed to do so. Machine learning is different to traditional programming in a number of respects.

Firstly, in traditional programming, specific instructions are written, whereas machine learning is different, in so far as they are learning different patterns from data, so the programmer isn't telling the system what to do. It is figuring it out itself by going over the data. This is useful when you have a large dataset. The machine learning model can go over the data and spot patterns, then make predictions of the patterns (IBM, 2025).

2.4.2 Types of Machine Learning Algorithms

Machine learning algorithms are grouped - by the different ways that they learn off of data. Examples of these different algorithms are 'supervised', 'unsupervised' and 'reinforcement' (IBM, 2025).

Supervised machine learning is when the model is specifically trained on a labelled data set. It is typically used in predictive analysis, image recognition and fraud detection. This is an effective type of machine learning as, as the data is clearly labelled, it becomes very accurate. When it is training it has the correct labels, which therefore allows ensures it has very good prediction rates (IBM, 2025).

Unsupervised machine learning models train on data that is unlabelled. This means it doesn't get any answers during its training and this allows the model to identify patterns on its own. Its common tasks would include pattern spotting and cluster analysis, which then allow people such as data analysts, to spot patterns within large datasets (IBM, 2025).

Reinforcement learning is when the model learns through a system of rewards and punishment. The model is learning by completing tasks and being rewarded for doing it in a specific way. When the model strays from this and attempts a different way, it is then penalised. This then helps the model in its decision making and optimising its accuracy over time (IBM, 2025).

2.4.3 Real-World Applications of Machine Learning

Machine learning is used widely across a multitude of industries, as it assists companies with decision making and working with data.

An example is the 'image classification model' which is made to detect images of cars and determine what type of car they are. This model can then be used for websites to stop spam bots. Recommendation engines on shopping websites are used to see the patterns of what shoppers . In this way such companies determine what to push as the most commonly purchased items by customers. 'Robotics use machine learning' is a way to train the robot on how to complete the tasks - the most optimal way to train the robot (IBM, 2025).

2.4.4 Machine Learning in Sports

Machine learning is being actively used more in sports. As machine learning helps understand large data sets this helps massively in sport. Machine learning is needed for its ability to track large amounts of data and to spot patterns in large amounts of data. Data analysts in sport use machine learning for things such as injury prevention. They do this by tracking the data of an individual player to see their injury history and try to spot the patterns that are leading to the injuries - something especially useful if a player is getting the same type of injury. Recruiting new players for a team is now almost impossible to do without machine learning. With machine learning, teams can now see

all the data on players and filter it down to see what players specialize in the areas that their team need ('Data and decisions in soccer,' 2022).

2.4.5 Challenges and Future Directions of Machine Learning in Sports

Even with the many advantages of machine learning in sport, there still remain many challenges that impact its effectiveness.

For example, machine learning needs accurate data for it to reach its full capabilities. A lot of data that comes from sport is retrieved from wearable sensors, cameras and tracking systems. If any of them become faulty or break, this can lead to poor data and so the prediction model can be incorrect. If a model is made and trained on one specific team, then biased data and prediction could also happen, making that information useless to other teams and readers.

Making the data readable and understandable to all readers and people who need to look at this data is imperative. Is something that is constantly happening, this would help all involved as most people in the sports world don't understand machine learning. So, making the outcomes of machine learning models accessible to everyone is of the up most priority (Perin *et al.*, 2018; Cairo *et al.*, 2020).

2.5 Technologies and Environments

2.5.1 Overview of Data Visualisation Technologies

There are many different types of technologies that are used to convert raw data and machine learning outputs into visualisations. Programming environments such as python are used to create these graphs. Visualisation libraries such as Plotly can also be used at the same time to help create better and more interactive visuals (Perin *et al.*, 2018; Cairo *et al.*, 2020).

2.5.2 Python as a Data Analysis and Machine Learning Tool

Python is one of the most common and widely used programming languages for machine learning. This is due to its readability, easy to use nature and host of usable tools. It works well with datasets that are large and complex which is useful for sports analytics. It supports every step of the workflow of data analysis - data processing, model development and it also has data visualisation tools that are easily integrated. This all allows data analysts to experiment and test out different models and libraries (Raschka *et al.*, 2020).

2.5.3 Dash and Plotly for Interactive Visualisation

Dash and Plotly are tools (Dash, 2024; Plotly Technologies Inc., 2024) that are used to help create interactive visualisations within Python. These tools can create interactive charts and are very useful as you can then get more in-depth information from the charts, if you wish to focus on specific piece of data. This is done by using interactive features like zooming, filtering and the hover tool. Dashboards are also good ways to display multiple visuals together, which becomes especially useful if you have created a few

different graphs around the same topic and want to display them together (Perin *et al.*, 2018; Cairo *et al.*, 2020).

2.5.4 Integrating Machine Learning with Visual Analytics

The reason we make these visuals from the machine learning data is to help everyone gain an understanding of the data. The data that typically comes from machine learning can be difficult for general readers to understand. Therefore, turning this data into graphs makes the data accessible to everyone, and turning this data into visuals, then makes spotting matters (such as patterns and trends) a lot easier for everyone, including the data analysts. Without the visuals, most people in sport would not be able to use the data that is gathered (Perin *et al.*, 2018; Cairo *et al.*, 2020).

2.5.5 Benefits of Interactive Dashboards in Sports Analytics

The benefit of making dashboards and graphs interactive in sports cannot be understated. It is true – having a graph that isn't interactive can still be of use to data analysts, coaches and players, as they still show important information. However, making them interactive adds much more information and allows for quicker and more informed decisions. Being able to filter through the data can help explain the data to the players for example, who would typically have no sense of what to make of all the data. Filtering the data to the point that you have only the specific parts that the player's need make the process much smoother. Sports remains a time sensitive business and decisions need to be made quickly and effectively. Interactive graphs therefore then allow all concerned to come arrive at solutions and decisions faster (Perin *et al.*, 2018; Cairo *et al.*, 2020).

3 Requirements Analysis

3.1 Requirements Gathering & Modelling

The requirements for this project were sourced through research, analysis of existing similar websites and consideration of the user's needs. Already existing football statistics and data platforms were also analysed to understand the most commonly used features. These include league tables, player statistics and team data. Harvesting this information helped identify the core requirements that users would expect.

In addition, the previous research section informed the best way to design the system - specifically in terms of how the data should be presented and making sure it was visually easy to read and understand. The importance of interactive dashboards and graphs was the key factor when defining the system requirements.

The users' needs were also considered. With the target audience being football fans, students, and anyone that is interested in sports data analysis, these type of users will require an intuitive and easy to understand work interface, which will allow them to explore and interpret the data with ease.

Overall therefore, the requirements were gathered to make sure that the website provides accurate data, which is clearly readable and understood, and that has interactive features that will enhance the user experience.

3.2 Functional Requirements

The functional requirements for the website are what determine the core features and operations that the website will need to perform. These requirements will describe what the website should do to meet the users' expectations and the project objectives.

- The website will retrieve data from an external API, such as fixtures, league standings, player data and team data.
- The website will display the football data in a clear and well-structured format across different and unique pages.
- The data that is displayed on the website should be shown thought clear, easy to understand and interactive graphs.
- The user will be able to filter through the data on criteria such as league, season, team and player.
- The website will allow comparison between teams and players using visual data.
- The website will update the data dynamically based on the users' iterations.
- It should handle API errors and then display the correct messages when certain data cannot be retrieved.
- The website should have loading messages when retrieving the data.
- The website should have a predictive feature that at least predicts upcoming matches.

3.3 Non-Functional Requirements

The non-functional requirements will describe the features that would ideally be added and implemented into the website but are not necessary for the website to run. These requirements will focus on things such as usability, performance reliability and the users overall experience.

- The website should work on all devices so computers, tablets and phones.
- The website should load the data efficiently and display the resulting charts in an acceptable time frame.
- The website should have a user-friendly interface and design, so it is easy to navigate
- The website should display consistent data from the API.
- The website should be able to handle all errors well and give clear feedback when an error does happen.
- The website should be structured to it can be updated in the future with ease.

3.4 User Personas

User are used to show the different type of persons that will be using this website. These personas will help guide the design and structure of the website by identifying the user's needs and goals.

Persona 1: Casual Football Fan

A casual football fan who wants to just have a quick look at league standings, fixture predictions and player stats. This type of user would tend to prefer simpler and easy to understand graphics. Would not require advanced analysis tools.

Persona 2: Football Analyst/ Student

This would be a more advanced user who would want to analyse football in greater detail. This user would want to have access to interactive charts, comparisons between teams and players and explore some trends with the data.

Persona 3: Regular User

This would be a footballs enthusiast who is a follower of the sport. This user would want a balance between what the two previous users would like. Simple information while also having deeper insights and access to features such as filtering and comparison charts.

3.5 User Cases

The user cases will describe the main type of interactions that the users will have with the website.

- A user can view football stats, league standings, upcoming fixture predictions, previous league standings and player and team stats
- A user can filter for a specific team and player
- A user can filter to previous seasons stats
- A user can compare a player and teams using visual data
- A user can view interactive charts to analyse the trends
- A user can use interactive features on charts such as hovering to get more info

4 System Design

4.1 System Architecture

The system architecture defines the overall structure of the website and how everything interacts. The website follows a web-based dashboard architecture. Football data is retrieved from an external API, processed within the system, and then presented through the interactive charts.

The frontend side of the website has been developed using Dash (Dash, 2024), Dash provides a structure and layout for the interface. The backend logic of the website was

developed using Python (Raschka *et al.*, 2020). This handled data processing, the API requests and the applications behaviour.

The actual football data is retrieved from API-Football (API-Football, 2024). This API provides real time football data such as league standings, upcoming fixtures, team and player stats and results. This data was then processed using the Pandas library (Pandas Development Team, 2024) just before being passed into visualisation components.

The interactive charts and graphs that were made were done so using Plotly (Plotly Technologies Inc., 2024). Using this allows the users to explore the data more dynamically through the use of filtering and interactive charts. The system architecture done ensures that the data flows efficiently from the API to the interface. This provides responsive and interactive experience to the user.

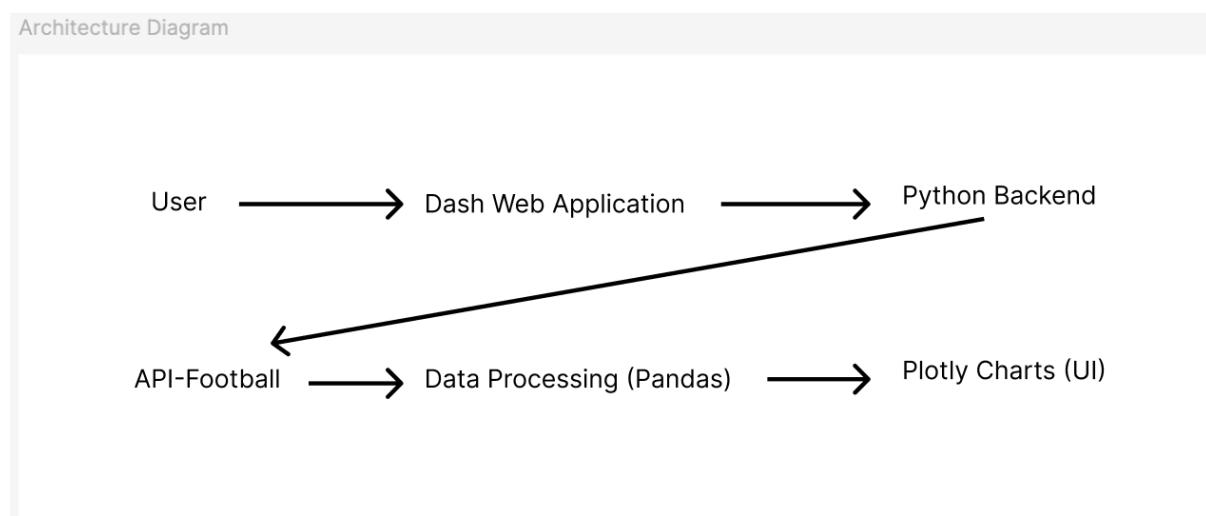


Figure 7- Architecture Diagram

4.2 Interface and User Experience Design

The focus for the interface and user experience design was to create a system that is visually clear, intuitive for the user and easy to navigate. The main goal of the design is that users can quickly access and understand football data without there being any unnecessary confusion or complexity.

The layout of the website follows a dashboard style, where the key information is presented on each page. The navigation of the website is structured to allow the users to move freely and easily throughout the different sections, which are leagues, teams, players, comparison and predictions. This allows the user to quickly find the information they want.

The main focus of the project and website was on the visuals be clear and easy to understand. The charts and graphs are designed to avoid clutter and difficult to read data, so the focus is to only display the necessary information. Simple visuals and charts such as bar charts, line chart and donut charts are used to compare the data and spot trends easily.

Interactivity is one of the key features that will determine a user's enjoyment while using the website and their experience. Filtering, searching and hovering are features that add to the users experience and allows them to explore the data in more detail. Having these features then allows the user to have a more engaging experience and to focus on specific info that they want to see. Overall, the design of the website aims to be simplistic and functional, thus providing both the causal and more advanced user's with an easy and effective way to explore football data.

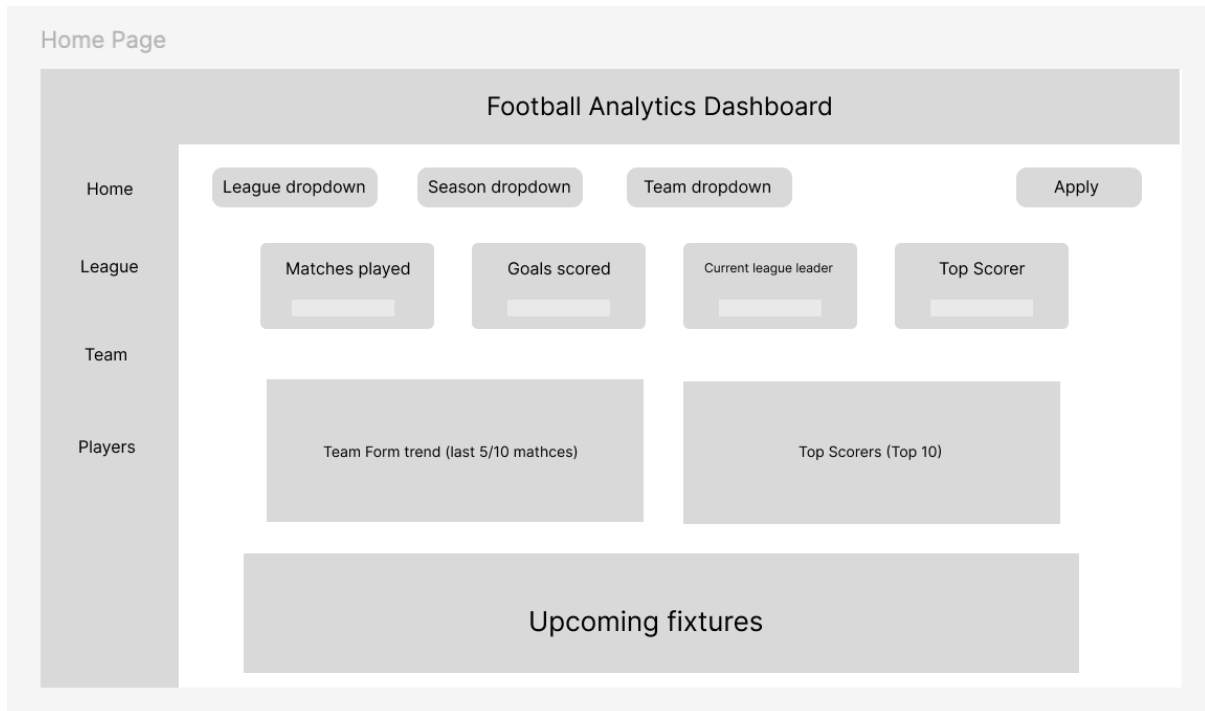


Figure 1- First mock up design of the home page

This was the mock up design of the home page. The interface allows the users quick and easy access to the data and easy navigation between the different sections of the website. The layout was designed for it to be simple and easy to navigate through.

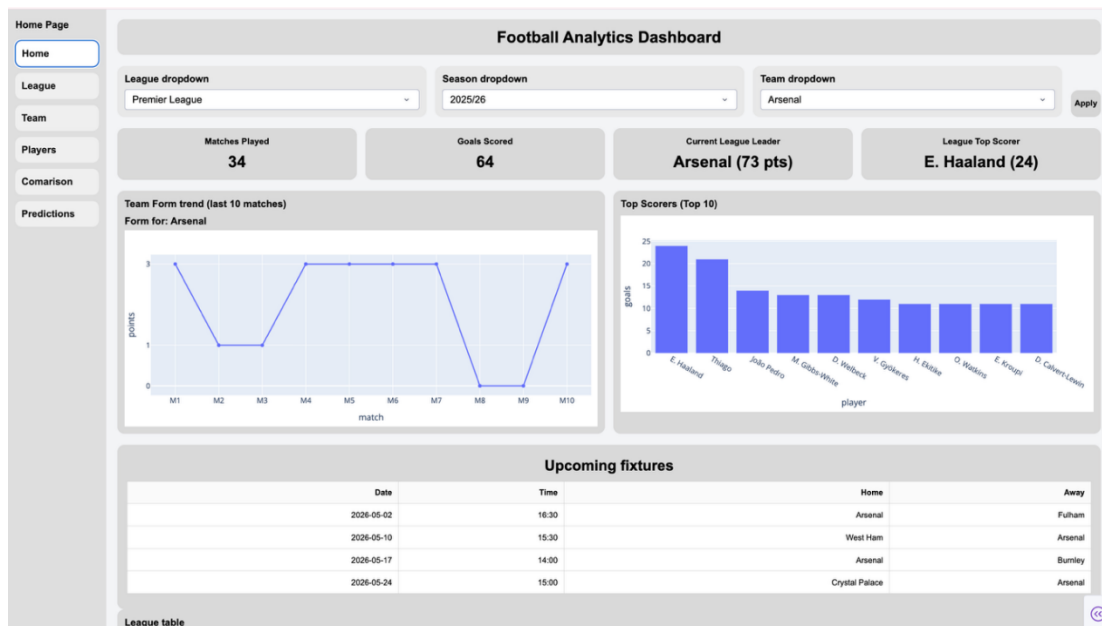


Figure 2 - What the Home page ended up looking like

This was what the final homepage looked like. This was required, as it is very similar to the first design mock up.

4.3 Data Visualisation Design

The data visualisation design is focused on presenting football data in a clear, easy to understand and interactive manner. The goals of the design were to transform the raw data from the API and convert it into visuals that are appealing and which will allow the user to quickly identify and understand patterns, comparisons and trends.

The different types of charts were picked based on the type of data that was been shown. Bar charts were selected when displaying data such as goals, assists and tackles between players in a team or league - they provide a clear and simple comparison. The line charts were chosen for stats that are best displayed over time, such as cumulative points. These charts were chosen as they are understood by most and easy to scan.

A key focus for the design was to not have cluttered data and ensure that each chart was clear with what it was trying to show. This was done to ensure that the readability is good and would prevent users from becoming overwhelmed with information. Clear labels and titles were therefore inserted throughout the website to make sure the charts were easier to read.

Making sure the charts were interactive was an important part of the design. The filtering, dropdown and hover features give the user the option to explore the data in more detail. The user can then focus on the specific data that they want, making the experience more meaningful and personalised.

Overall, the aim for the data visualisation was to represent the data in a clear and easy to read format, while also having good interactivity options. This makes the website great for casual and also, more experienced users.

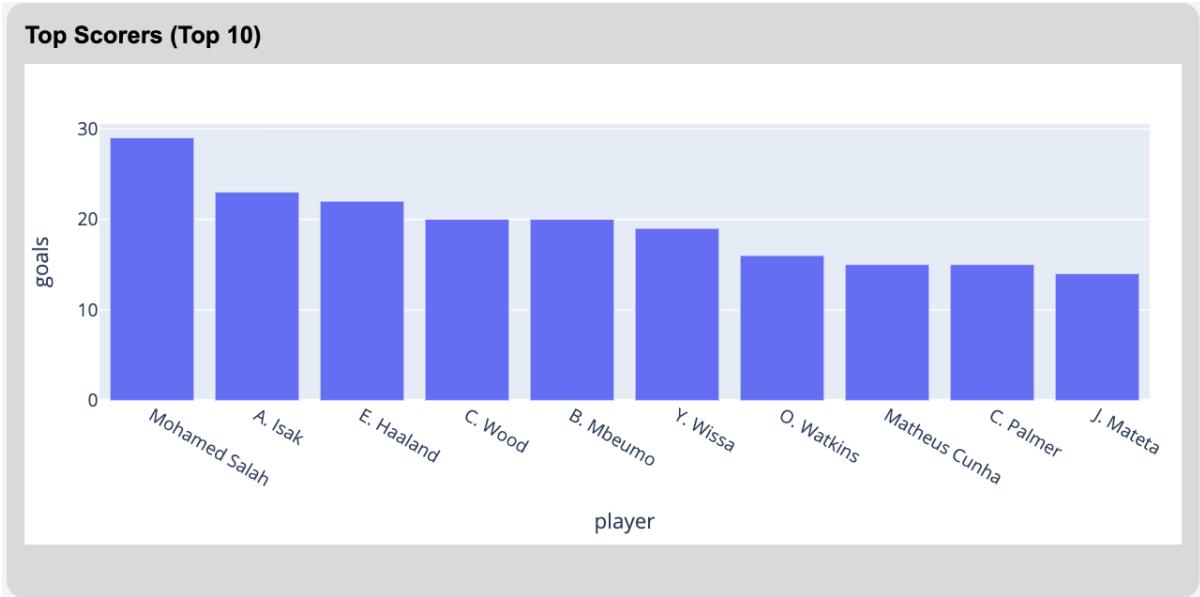


Figure 3 - Top Scorers Bar chart

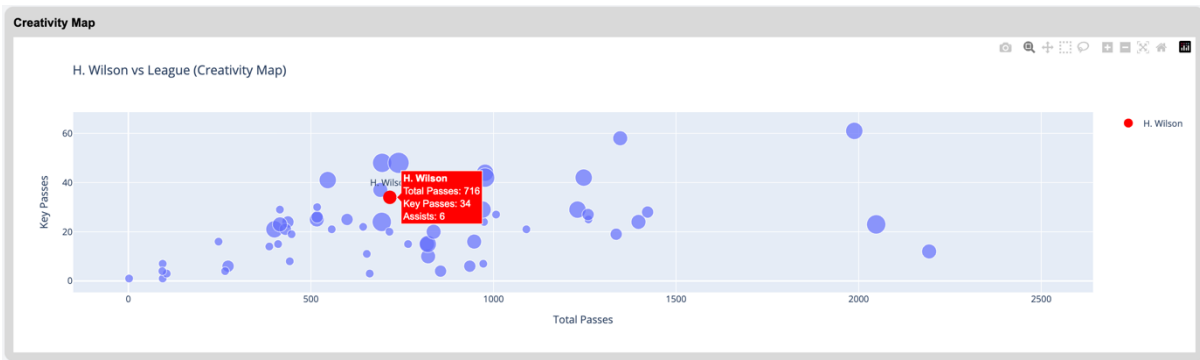


Figure 4 - Bubble chart showing the hover feature

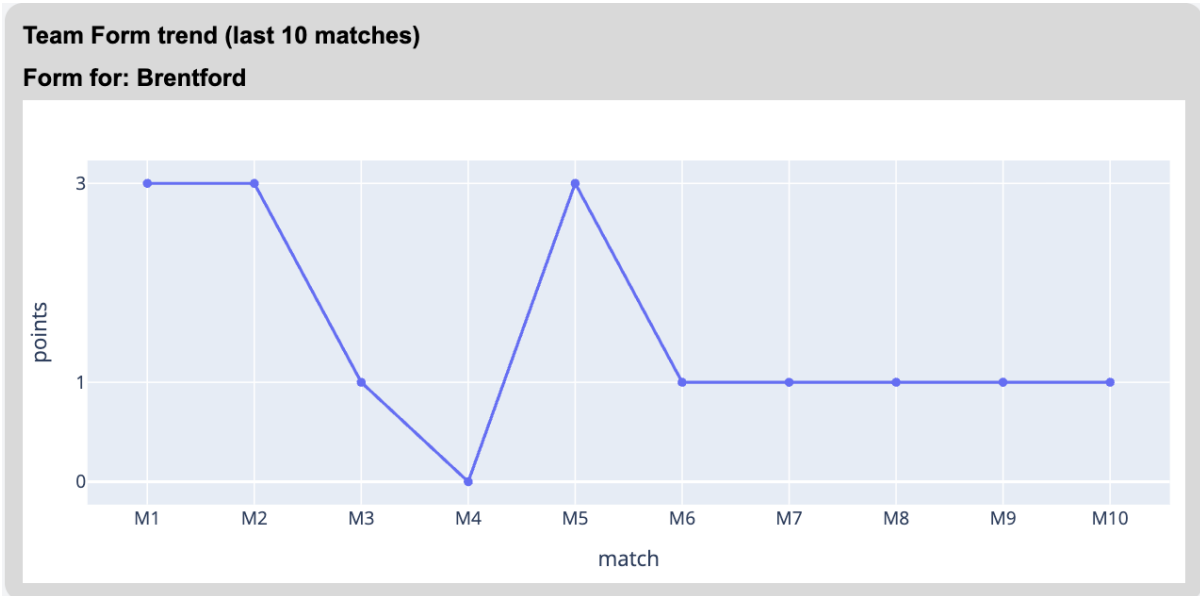


Figure 5 - Showing the different types of charts, this is the line chart for the team form

4.4 Database Design

The database for the website differs from a traditional one as it doesn't store any data locally. Instead, it gets all the data from an external API. The website uses API-Football as the main data source, the API provides well-structured football data in JSON format. This info includes fixtures, league standings, player and team stats.

Once I have retrieved the data, it was then processed and structured within the website using Python and the Panda's library. This allowed the website to organise the data into table so it could easily be formatted into the visuals.

The actual structure of the data is based on key parts such as leagues, teams, player and match results. These pieces of data are linked through identifiers, which allows the system to display the relevant info based on the users input.

Therefore, as the website does not store persistent data, it thus avoids the need for database management systems. However, the approach to the data handling ensures that information is processed efficiently and is ready to be used with the interactive visuals.

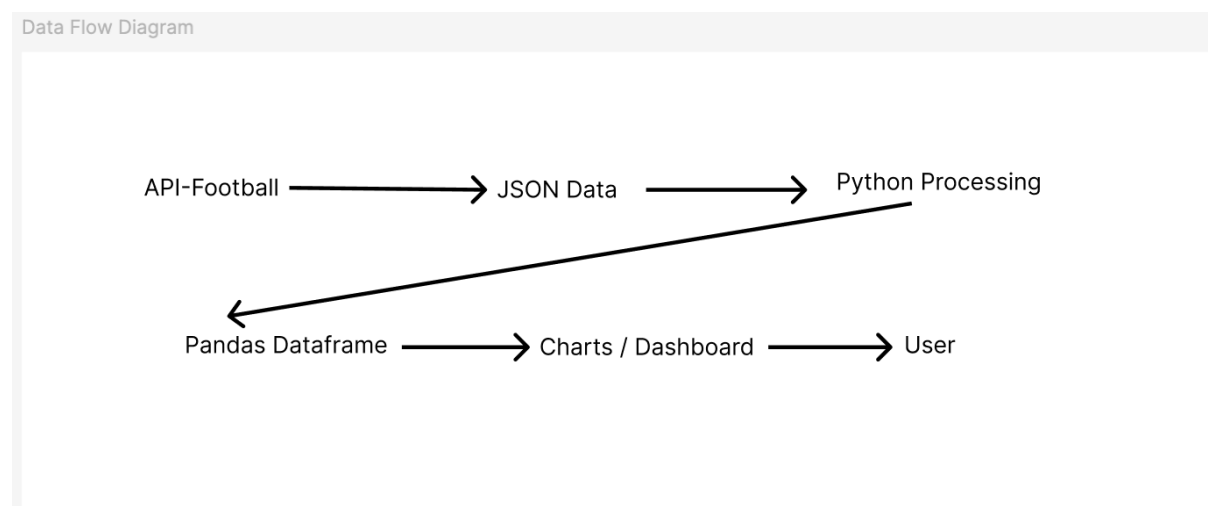


Figure 6 - Data Flow Diagram

5 Implementation

5.1 Development Methodology

An iterative approach was taken to develop this project. This means that the project was built in stages, which allowed for continuous improvement and testing throughout the whole development process.

The first steps were to set up the development environment and to integrate the external API that retrieved the football data. When this was completed, the next step was to start developing the user interface and implementing the data visualisation features. The additional functionality features such as filtering, hovering and searching were then added and worked on over time.

This approach allowed for more flexibility during the development, which made it possible to identify and fix the issues early on, while working on the website simultaneously. This method also made sure that each component of the website was fully working before moving on.

5.2 Frontend and Backend Development

The Dash framework and Python were implemented to create the frontend and backend of the website. Dash was used to create the frontend user interface, Python was used to create and handle the backend logic. This included the data processing and the interacting with the External API.

The focus of the frontend of the website was on presenting the football data in a clear and interactive. The design of the layout was done through Dash components, with separate pages for different areas of stats such as a league, team, player, comparison, predictions and home page. A sidebar was made to ensure page navigation was easy and efficient. The interface was designed to be user friendly and to ensure that the user can quickly understand the flow of the website and how to access the data that they require.

The backend of the website is what gathers the data from the external API - it retrieved the data and then processed it. API requests are made to API-Football to gather the real-time data such as fixtures, results and players/team's statistics. This data was then processed using Python and the Pandas library to make sure that is structured in the right way before being displayed.

The interactivity within the website is done using Dash callbacks. These callbacks is what allow the system to respond dynamically to the users input such as, filtering, selecting a player or team and updating the charts. This is what ensures that the website remains responsive and interactive.

AI tools were used in some parts of the development process to assist with debugging and some problem solving. Tools such as GitHub Copilot and ChatGPT (GitHub, Inc., 2024; OpenAI, 2024) were used to help identify errors, suggest possible solutions, and improving the code structure. In particular, it helped with Dash callbacks and the handling of API data. All the suggestions that were given were reviewed and tested to make sure that they worked correctly and that they fitted in with the code structure before being implemented.

Overall, the combination of the frontend and back end is what allows the website to be efficient in retrieving, processing and displaying the football data. Providing the user with a smooth and enjoyable experience on the website.

```
@dash.callback(  
    Output("player_stats_chart", "figure"),  
    Output("player_radar_chart", "figure"),  
    Output("player_passing_chart", "figure"),  
    Output("player_scatter_chart", "figure"),  
    Output("player_donut_chart", "figure"),  
    Input("player_apply", "n_clicks"),  
    State("player_league_dd", "value"),  
    State("player_season_dd", "value"),  
    State("player_team_dd", "value"),  
    State("player_player_dd", "value"),  
)
```

Figure 11 - Dash callback to update player visualisations

This code shows the callbacks that were used to update the multiple player charts at the same time. When the user selects a league, season, team and player the callback updates all the charts dynamically.

```
df = pd.DataFrame([{"stat": k.capitalize(), "value": v} for k, v in stats.items() if k != "passes"])  
  
fig = px.bar(  
    df,  
    x="stat",  
    y="value",  
    title=f"{player_name} Key Stats ({season_year})",  
    text="value",  
)  
fig.update_traces(marker_color="steelblue", textposition="outside")  
fig.update_layout(yaxis_title="Value", xaxis_title="")
```

Figure 12 - Plotly bar chart for player key stats

This is the code that converts the selected player stats into a DataFrame and then creates a bar chart using Plotly. This chart displays the players key stats in a clear and easy to read format.

```
# I convert the rows into a Dash DataTable for display
table = dash_table.DataTable(
    columns=[{"name": c, "id": c} for c in rows[0].keys()],
    data=rows,
    page_size=10,
    style_table={"overflowX": "auto"},
    style_cell={"fontFamily": "Arial", "fontSize": 13, "padding": "10px"},
    style_header={"fontWeight": "bold"},
)
```

Figure 13 - Dash DataTable to display prediction results

This is the code that converts the prediction data into a Dash DataTable. So, fixture predictions are allowed to be displayed in a structured and readable manner for the user.

5.3 Data Integration and API's

The website integrates the data from an external API, which provides real-time up to date football statistics. API-Football is the API that was used to do this, this is because it provides structured data which include fixtures, results, league standings, player stats and team information.

The API requests are made using Python, specific endpoints are called to retrieve the data that is required. The responses are then returned in a JSON format, which then get processed and structured within the website. This data that is retrieved is then cleaned and organised using the Pandas library. This is done to ensure that the data is in a suitable format for visualising and displaying within the website.

Error handling was implemented into the system to improve the reliability. This helped manage issue such as failed API requests or missing data. Loading indicators and messages were then added to keep the user informed when the data is being retrieved.

The integration of the API is what allows the website to provide real time and relevant football data. Which is what is essential to delivering the accurate and interactive visuals.

```

def api_get(path: str, params: dict) -> dict:
    key = _make_cache_key(path, params)
    now = time.time()

    if key in _API_CACHE:
        saved_ts, saved_data = _API_CACHE[key]
        if (now - saved_ts) < CACHE_TTL_SECONDS:
            return saved_data
        else:
            del _API_CACHE[key]

    url = f"{BASE_URL}{path}"
    r = requests.get(url, headers=api_headers(), params=params, timeout=25)

```

Figure 8 - API Request and caching function

The code here shows how the system is retrieving the data from API-Football. It is also using caching so the repeated requests can return saved data and not calling the API every time, this helps the performance.

```

def get_league_top_scorers(league_id: int, season_year: int) -> pd.DataFrame:
    data = api_get("/players/topscorers", {"league": league_id, "season": season_year})

    rows = []
    for item in data.get("response", []):
        player = item.get("player", {}) or {}
        stats = (item.get("statistics") or [{}])[0] or {}
        goals = (stats.get("goals") or {}).get("total", 0)

        rows.append(
            {
                "player": player.get("name", "Unknown"),
                "goals": int(goals or 0),
            }
        )

    df = pd.DataFrame(rows)

```

Figure 9 - Processing top scorer data

The code here retrieves the top scorer data from the API and then converts the response into a Pandas DataFrame. This then makes it easier to sort, filter and pass into the charts.

```

def get_league_table(league_id: int, season_year: int) -> pd.DataFrame:
    """Return a league standings table for the given league/season.

    The returned DataFrame includes a hidden `team_id` column so that
    Dash can highlight the selected team row without exposing the ID in the UI.
    """

    data = api_get("/standings", {"league": league_id, "season": season_year})
    resp = data.get("response", [])
    if not resp:
        return pd.DataFrame()

    league = (resp[0] or {}).get("league") or {}
    standings_lists = league.get("standings") or []
    if not standings_lists or not standings_lists[0]:
        return pd.DataFrame()

    rows = []
    for row in standings_lists[0]:
        team = row.get("team") or {}
        all_stats = row.get("all") or {}
        goals = all_stats.get("goals") or {}

        rows.append(
            {
                "rank": row.get("rank"),
                "team": team.get("name"),
                "team_id": team.get("id"),
                "played": all_stats.get("played"),
                "win": all_stats.get("win"),
                "draw": all_stats.get("draw"),
                "lose": all_stats.get("lose"),
                "goals_for": goals.get("for"),
                "goals_against": goals.get("against"),
                "goal_diff": row.get("goalsDiff"),
                "points": row.get("points"),
            }
        )

    df = pd.DataFrame(rows)
    if df.empty:
        return df

    return df.sort_values("rank").reset_index(drop=True)

```

Figure 10 - League table data structure

This code shows how the league table is structured. The returned data is organised into different fields such as ranks, team, wins, draws, losses and points so it can then be displayed clearly.

```

# I fetch the next 10 upcoming fixtures for the selected league/season
fixtures = get_league_upcoming_fixtures(league_id, season_year, next_n=10)
# If no fixtures are found, I return a message
if not fixtures:
    return html.Div("No upcoming fixtures found."), ""

rows = []
# For each fixture, I request prediction data from the API and build a row for the table
for fx in fixtures:
    fixture_id = fx.get("fixture_id")
    # I call the prediction endpoint using the fixture ID
    pred = get_match_prediction(fixture_id) if fixture_id else {}
    # I extract the relevant prediction details, handling cases where data might be missing
    predictions = pred.get("predictions", {}) if pred else {}
    winner = (predictions.get("winner") or {}).get("name", "No prediction")
    advice = predictions.get("advice", "No advice")
    percent = predictions.get("percent", {}) if pred else {}

```

Figure 14 - retrieving prediction data from API-Football

The code here retrieves the upcoming fixtures and then calls the API-Football prediction endpoint for each of the fixtures. The website does not generate its own prediction model, instead it displays API-Footballs prediction model.

5.4 Challenges and solutions

While developing the project, there were several challenges that were encountered across different parts of the website. One of the major challenges was having to manage many different API calls across the different pages on the website. Each of the pages required different types of data such as league standings, fixtures, player/team statistics and predictions. This issue was then fixed by separating the API logic into a different file and creating reusable functions that handle the different API Endpoints. This improved the structure of the code, making it easier to maintain.

Another challenge was dealing with the raw data that was returned from the API. The data was returned in a JSON format, which didn't always work for everything on the website. This made it difficult at times to display the data in an effective way. To fix this issue, the Pandas library was used to clean, structure and organise the data before it was passed into visualisations.

There were also difficulties while implementing the dynamic interactivity using Dash callbacks. Sometimes the callbacks would return an incorrect number of outputs which then led to errors such as schema validation issues. This was fixed by carefully checking all the callback statements then ensuring that they matched the number of outputs.

There was also an issue with the dropdown menus - specifically when dealing with the teams and players. The initial approach had the dropdowns using fixed data, this caused inconsistencies though when users tried to change league or season. Callbacks were implemented that dynamically updates the dropdowns options based on the user's choices. This ensures that only the relevant data is displayed.

The data visualisation in general caused some issues. In particular, when large datasets were involved and keeping charts clean and easy to read. Some of the charts initially appeared cluttered or large pieces of data made the graphs look wrong. This was fixed by selecting the appropriate chart types, simplifying the data and ensuring the hover and labels were improved.

The design of the prediction page also posed an issue. When first made the user had to select and home and away team for the prediction. This didn't always work as the prediction logic is based on upcoming games, so if two teams already played twice nothing came up. This was changed to show the predictions of the next round of fixtures in the users chosen league.

All these challenges were resolved through testing and debugging resulting in a more stable, accurate and user friendly website.

```
radar_fig.add_trace(  
    go.Scatterpolar(  
        r=player_values_closed,  
        theta=categories_closed,  
        fill="toself",  
        name=player_name,  
        customdata=player_raw_values_closed,  
        hovertemplate="%{theta}: %{customdata}<extra></extra>",  
    )  
)
```

Figure 15 – Radar chart hover

This code was used to fix an issue where the hover didn't display the stats of a player that wasn't on the chart by default. The original raw values are passed through customdata, so users see the correct stats when hovering.

6 Testing and Evaluation

6.1 Test Plan

Functional user testing will be done to make sure that the website is working as intended. Three people will complete some tasks on the website, they will then document what they thought worked and didn't work from the website. These tasks will be a broad range and will have the tester explore all parts of the website and test out the many different features.

These include navigating the different pages, using the correct filters and find the stats that they are being asked to find. This ensures that all key features of the website are being tested.

The results of the tests will be recorded using test tables – this includes the instructions of the tasks, the expected results, if the users passed or failed the tasks and comments from the tester. This gives clear user feedback on what to improve in the future.

6.2 Testing Results

The results from the user testing showed that the websites feature functioned correctly and did as they were intended to. All of the tasks were completed by the three participants and without much difficulty.

The website was able to retrieve the data from the API and display the data without any issues. The interactive elements like the charts and filters did what they were meant to do. Overall, the functionality of the website met the expectations.

The testers still had some feedback on some parts that they thought could be improved. The first two testers said that they liked the layout of the dashboard but also thought that a bit more colour to the website would help. They also thought that adding some pictures of the players and the club's logos would help for people who don't know much about football.

The other tester like the majority of the website, but they felt that the filters could have been laid out more cleanly on some pages, such as the comparison page. They also thought some of the graphs could be a bit better explained, such as the points distribution graph.

Overall, the testers liked the website, and it all functioned accordingly. At the same time though there were certain things that could be improved, but nothing overly detrimental to the functionality of the website.

6.3 Usability Testing

Usability testing was completed to evaluate how easy or difficult the website was to use and how effectively a user could interact with the system. Testers were asked to complete a wide range of tasks which included navigating between the different pages, viewing different player and team statistics and using the filters to switch between leagues, teams, seasons and players.

The results and feedback show that the website is generally easy to navigate with the tests completing all the tasks that they were asked to do. The layout of the dashboard was liked and one of the highlights, with some of the testers saying that the layout made it easy to locate the data.

However, there were still some areas of improvement noted. Testers suggested that more use of colour throughout the website would make it look that bit more appealing. It was also recommended that adding images of the players and team logos would be useful for people who aren't that familiar with football.

Other users also noted that the filters on some pages such as the comparison page could be laid out a bit better. Additionally, it was noted that some of the graphs were not as clearly labelled and more difficult to read and understand the data than others. This was common with the points distribution chart.

The usability testing went well overall as the website had positive feedback overall. While most of the critiques were well noted and would be the first set of things to be improved on with the continued development of the website.

6.4 System Evaluation

The aim of this project was to develop an interactive football statistics website that had a focus on data visualisation and allowing the user to easily explore the data in an effortless and personal manner. Overall, the website meets this aim, as it retrieves real time data from the external API and displays the data through interactive charts and visuals.

All the core functions of the website work as intended, such as the league tables, player and team statistics and the comparison tool. Users can also easily switch through all the data via the filters on each page, while also being able to get more info from the charts using the interactive hover feature through the use of Plotly.

The testing results also support this All testers were able to complete each task that they were given and the website performed well during this. The feedback from the testers concluded that the website was easy to use and had a clear layout with an intuitive design.

However, as mentioned, and notwithstanding the above, there were still some areas that could be improved. For example, testers thought that adding a more colour to the website would make it look a bit more intriguing; also they also said that adding images of the players and club logos would be useful for people who don't know a lot about football. There were also some thoughts that the filters could be laid out more clearly.

An additional matter is that since the website relies on an external API, there can sometimes be delays in the data displaying. This may impact users in certain instances.

Overall however, the project can be considered successful, as it has met the main objectives set out at the start of the development progress. In other words, it provides a functional website to explore interactive football statistics.

7 Project Management

7.1 Project Planning and Organisation

The project was managed and organised in a structured manner that helped keep a steady progress. Tasks were broken up and done incrementally; the focus was on completing the pages by importance and difficulty.

The development of the project followed a logical approach beginning with research then design of the system itself, Implementation and testing then followed. This method helped with making sure that each stage was complete before moving onto the next.

Another important part of the project planning and organisation was time management, as the project was completed within an academic timeframe. The majority of the time was given to the most important matters to get completed early, such as API integration, data visualisation and the user interface. This was to ensure that the core functionality of the website was implemented.

Although a formal project management system was not used, the project was still completed in an organised and efficient way. With each page being worked on at a time, this ensured the most difficult features of each page were completed first before refining the functionality of each.

7.2 Tools and Workflow

A wide range of tools were used to help with the development of the website. Python was used as the main programming language, with Dash framework being utilised to make the user interface, whilst Plotly made the visualisations.

Visual Studio Code (Microsoft, 2024) and Git (Git, 2024) were used as the development environment. These provided handy features such as code editing, debugging and extension support – all of which made for an efficient development and testing of the website.

Git was used for the version control. Using Git allowed for the changes to be tracked throughout the development. GitHub (GitHub, Inc., 2024) was then used to store the projects repository and maintain the different stages of the code. This was the best way to track both the development of the project and the changes. It also allowed for the changes to be reverted if needed.

The workflow consisted of developing features of the website incrementally, testing them as they were implemented, then refining them depending on how they worked. The code was organised into separate files and folders such as pages, assets and services folder. This was to improve the readability and to keep everything organised.

The use of these tools overall helped managing the workflow for the development of the website. It helped ensure that the website was developed efficiently and in an organised manner.

8 Conclusion And Future Work

8.1 Summary of Findings

The project aims were to develop an interactive football statistics website that had its main focus on the visual side of displaying data. Overall, the website met this aim and was successfully developed, meeting the primary objectives. The website retrieves real time football data from the external API and is then presented through clear interactive charts and visuals.

The key features - including getting the league standings, player and team statistics and the comparison tool - were all implemented successfully and worked as they were intended too. The testing that was carried out showed that the testers were able to carry out the tasks without any major issues and navigate the website with ease.

This project shows how data visualisation can be used to make complex football data be displayed in an easy to read and understandable way.

8.2 Limitations

The website overall was a success but even with that there were still some limitations, one main limitation is the reliance and use of the external API for the data. This is great as it allows access to real time data – however that also means that the website is dependent on the availability, speed and structure of the API. This can then result in delays in the data loading and limits to the data that is able to be displayed.

There were also limitations with the interface, even though the layout is functional and easy to navigate. There are still some areas that can be improved that would enhance the overall experience for the user. An example would be the layout of the filters on certain pages such as the comparison page. The filters on this page aren't as neat as they should be. Some of the charts could also give a clearer explanation on what data it is displaying, such as the points distribution chart.

Additionally, the visual design of the dashboards and overall website is quite basic. There are also a lack of some visual elements such as the player images and team logos that were mentioned by the testers. Not having those elements does reduce the overall visual appeal and could make the website less appealing to users.

8.3 Future Improvements

There are several areas that I would like to improve in the future. One of the main areas of improvement is the visual design of the website. Adding more colour alongside the images of players and clubs would make the interface a lot more engaging. It would also help users who are unfamiliar with football.

Another improvement that I would want to get in the early stage of development, is the layout improvement of the filters. Overall, the filters are satisfactory but on certain pages they could be improved. For example, on the comparison page they are off centre due to

having two rows and the apply button. Additionally, I would also like to improve the look and feel of the apply button on all pages as I feel it is too small.

Improvements to the labelling on some of the more complex graphs is also something that could be improved. For the majority of graphs, they are easy to understand and gather what the data is displaying pretty quickly, but for some it takes a bit longer to understand. This would be an important improvement.

The comparison feature for the players works very well and displays good insightful data and improvement to this page would be made by also allowing the user to compare teams - as they have a lot of data that can be compared. This would add another level of insightful data to enjoy.

One final comment regarding improvements is regarding the prediction page and model. Currently I am using the built-in model from the external API. Ideally I would make my own custom one that can predict more than just the upcoming fixtures. Therefore, the model would also predict top scorers for the next season, the leagues standing and much more.

9 References

Cairo, A. (2020), *The truthful art: Data, charts, and maps for communication*. New Riders.

<https://www.peachpit.com/store/truthful-art-data-charts-and-maps-for-communication-9780321934076>

IBM. (2025), *What is data visualization?* IBM.

<https://www.ibm.com/think/topics/data-visualization>

IBM. (2025), *What is machine learning?* IBM.

<https://www.ibm.com/think/topics/machine-learning>

IBM. (2025), *Types of machine learning*. IBM.

<https://www.ibm.com/think/topics/machine-learning-types>

IBM. (2025), *Supervised learning*. IBM.

<https://www.ibm.com/think/topics/supervised-learning>

IBM. (2025), *Unsupervised learning*. IBM.

<https://www.ibm.com/think/topics/unsupervised-learning>

IBM. (2025), *Reinforcement learning*. IBM.

<https://www.ibm.com/think/topics/reinforcement-learning>

Perin, C., Vuillemot, R., Stolper, C. D., Stasko, J. T., & Wood, J. (2018). Criteria for rigor in visualization design study. *IEEE Computer Graphics and Applications*, 38(3), 87–97.

https://openaccess.city.ac.uk/id/eprint/19857/8/2018_eurovis_sportsstar.pdf

Raschka, S., Patterson, J., & Nolet, C. (2020), Machine learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, 11(4), 193.

<https://www.mdpi.com/2078-2489/11/4/193>

RudderStack. (2024), *The difference between data analytics and data visualization*. RudderStack.

<https://www.rudderstack.com/learn/data-analytics/the-difference-between-data-analytics-and-data-visualization/>

Strategy+Business. (2022), *Data and decisions in soccer*.

Strategy+Business.

<https://www.strategy-business.com/blog/Data-and-decisions-in-soccer>

Google Developers. (2024), *Linear regression*. Google.

<https://developers.google.com/machine-learning/crash-course/linear-regression>

Microsoft. (2024), *What is machine learning?* Microsoft.

<https://www.microsoft.com/en-us/microsoft-copilot/for-individuals/do-more-with-ai/general-ai/what-is-machine-learning>

Few, S. (2020), *Data visualization: Past, present, and future*. IBM Cognos White Paper.

https://public.dhe.ibm.com/software/data/sw-library/cognos/pdfs/whitepapers/wp_data_visualization_past_present_future.pdf

Du, Y., & Yuan, X. (2020), A survey of visualization techniques for sports data, *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 871–891.

<https://link.springer.com/article/10.1007/s12650-020-00687-2>

Dash. (2024), *Dash documentation*

<https://dash.plotly.com/>

Plotly Technologies Inc. (2024), *Plotly Python graphing library*.

<https://plotly.com/python/>

Pandas Development Team. (2024), *pandas documentation*.

<https://pandas.pydata.org/>

API-Football. (2024), *API-Football documentation*.

<https://www.api-football.com/>

Git. (2024), *Git documentation*.

<https://git-scm.com/docs>

GitHub, Inc. (2024), *GitHub platform*.

<https://github.com/>

GitHub, Inc. (2024), *GitHub Copilot*.

<https://github.com/features/copilot>

OpenAI. (2024), *ChatGPT*.

<https://chat.openai.com/>

10 Appendices

Additional materials including supporting files are available in the OneDrive folder.