



CS353 TEAM PROJECT

GROUP 16



**FOOTBALL STATISTICS
APPLICATION**

**Mohamed Ibrahim
20460384**

Team 16 Members – Bartek Kowal, Ismael Plangca, Conor Phelan, Jamie Hoban, Mohamed Ibrahim

CHAPTER 0 (2000 words)

The theme for this project was to build a web application, preferably backed by a server or database. We were to use the agile process, '**Scrum**' to perform this task. The goal behind using Scrum is 'to deliver the highest business value in the shortest time' (Casey K. 2022).

Scrum is an agile process, 'Agile' is a software development model which is based on iterative development. The application should be built to a basic extent first, and then added onto to make it more *sophisticated*. This system allows us to always have a working application where the improvements are therefore visible. For example, it is more satisfactory to design a viable home screen for the website before putting extensive work on other pages on the web application. It is harder to understand progress if we must wait until every detail of the project is completed before we have a functioning product.

Agile History

The idea of agile process models is to be versatile and adaptable to changes. In 2001, The '**Agile Manifesto**' was formally launched. The manifesto has 4 key principles which should improve teams developing software. The first principle is '**Individuals and interactions over processes and tools**', this means that the process is predicated on being personable, people must work together to complete tasks. If individuals work alone, strictly following procedures, the work can often take longer and be more complicated than if multiple heads work together to achieve progress. The second principle states '**Working software over comprehensive documentation**'. The idea behind this is that the development should be about showing visible progress rather than extensive description of the work completed. In the real world, the 'Agile Manifesto' believes that we should let our work do the talking, however as our project is being graded partially on this document, the process is slightly altered. The different aspect of a university project compared to a workplace product is also highlighted in the third principle of the manifesto; '**Customer collaboration over contract negotiation**'. As we do not have a customer or a contract for that matter, this is less relevant to the project. For us, the 'contract' is the list of rules set down by the lecturer and we must collaborate with the demonstrators and lecturer about the project as if they are the customer. However, we have much more freedom to make decisions about the product itself than if there was a contract. The final principle of the manifesto is '**Responding to change over following a plan**'. This reiterates the ideas of versatility and adaptability. While plans are important to get a project in motion, we must understand that obstacles occur, situations change, and our initial goals may transpire to be unachievable. Our responses to change can often turn into our greatest ideas because we have a greater understanding of the product during the process compared to when we make our initial plans.

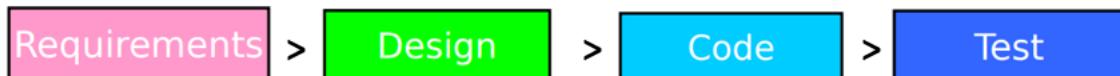
Scrum History

'Scrum' is a term which originates from the sport, rugby. In rugby, a scrum is two sets of 8 players, who are binding together as a unit, pushing against the other unit of 8. The software development usage of 'Scrum' relates to this unit of 8 people working together as a team. The term was first used in a software development context in 1986 by Hirotaka Takeuchi and Ikujiro Nonaka in their paper, 'The New New Product Development Game'. Having reviewed some of the paper, some of their rugby knowledge leaves a bit to be desired, however, their software development ideas are clearly both intelligent and revolutionary.

Since its first use in the 1990s, Scrum has been used by all top companies involved in software development. Scrum has also been used for many different projects by these companies including video game development, financial applications and mobile phone development.

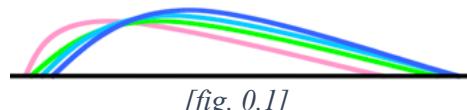
Scrum Breakdown

Once a project is agreed upon, one major process involved in Scrum is the creation of a product backlog, which is a list of tasks we need to complete the building of the product. This product backlog is created after we go through what the users of this product need or want. We must prioritise different tasks and allocate them to different stages of the product development. This is where another crucial component of the Scrum process comes into play. The process is broken up into subsections called '**Sprints**'. In the real-world, the sprints would be roughly one month long. Much like the project as a whole, we have a Sprint backlog, this is the list of tasks we have decided should be completed over the next stage of the project. While we can estimate which stage of the process tasks will be completed in, we will encounter difficulties which may change our initial estimations. This means that restructure, especially of the Sprint backlog, is important when we are transitioning from one sprint to another. In non-agile processes, the tasks were usually completely in a sequential basis (fig. 0.0).



Sequential Image [fig. 0.0]

In Scrum, we try to do parts of each of these at the same time to ensure we always have a functioning project. Unlike the sequential order, it follows an overlapping system (diagram below with each colour consistent with the above diagram).



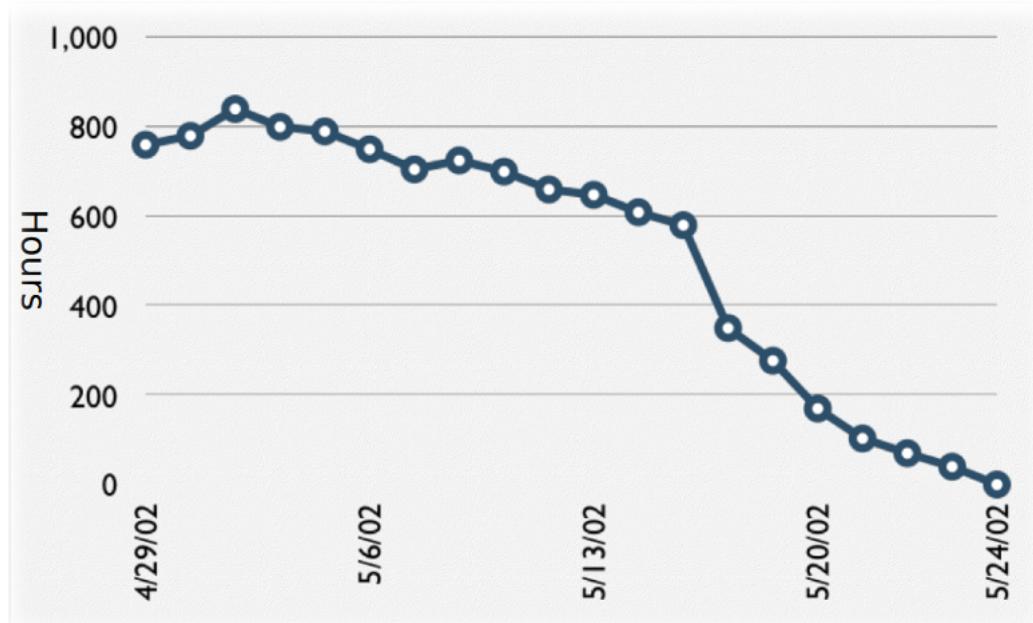
[fig. 0.1]

As we are only capable of meeting once or twice weekly for our project, we did not have **daily scrum meetings**. However, in real world companies, this is common. The idea of daily meetings is to give regular updates of progress and issues. The idea of this meeting is not to solve problems we have encountered in the development, it is to ensure everyone is up to speed on what each person in the group is doing. Ideally, we discuss what each person did yesterday, what each person will do today and what issues you have encountered along the way. It should not be a long meeting; the ideal length is roughly 15 minutes. The scrum master* (more detail below) should take charge of the meeting, ensuring everyone gets speaking time and that the meeting is being used appropriately.

There are three other types of Scrum meetings: **Sprint Planning**, **Sprint Review** and **Sprint Retrospective**. These are all rarer than the daily meetings and occur based on how many sprints we have across our project duration. The sprint review is a meeting where all the team presents the work completed during the sprint and the product we have so far. This is an informal meeting which means the team should not spend an excessive amount of time preparing the material to be presented. The sprint review is primarily about the product development not about the scrum process. In the sprint retrospective meeting is where we discuss what parts of the scrum process worked and did not work. The theme of the

meeting revolves around the START, STOP, CONTINUE model. This allows all team members to give their honest assessment of the previous sprint and give their input on what needs to be done, what needs to be stopped and what worked well. The sprint planning meeting is where we set goals for the upcoming sprint. We try to form an updated product backlog and which parts of that backlog are specific to the sprint we are beginning (**sprint backlog**). In the Scrum ideology, group members should not be assigned different tasks, people should instead be encouraged to volunteer their services to tasks they feel they can complete. This can lead to certain members doing more than others at first but if there is honesty in sprint review meetings and sprint retrospective meetings, we should achieve an equal input level.

The group also attempts to estimate the length each task will take. The nature of these estimates is that they are tough to get correct and some group members have a better grasp of the time that will need to be put into different tasks. This estimation of the labour hours required to create our product leads us onto the creation of '**Burndown Charts**'. There are multiple methods of estimating the time tasks will take. As people with little understanding of a task may give outlandish estimates, we could take the average of everyone's estimates to get a fairer guess of the task time. This is often described as '**Planning Poker**'. After each task time is estimated, we will get a total amount of time (usually in hours) that we expect the task to take. We expect that this will diminish as the project goes on. It may not decrease in a linear fashion and at certain points the estimated time may increase if we have underestimated tasks in our initial plans. Below is an example (from the Week 1 slides) of how a burndown chart may go:



[fig. 0.2]

Team

In the real world there would be a **product owner**. This is not really a factor in our project but in reality, this person would be dictating the release date of the product and is responsible for ensuring the product is profitable. For us, there is no financial aspect of the project, this may have positives in terms of freedom and negatives as funding lessens certain obstacles.

While businesses have management hierarchy procedures, in the scrum process, the team organises itself and has a '**Scrum master**' to take charge of the meetings. The self-organisation aspect is a fundamental part of Scrum, and these teams tend to be made up of between 5-9 people. The Scrum master represents the management in a process which is separated from management. The Scrum master is responsible for

the project being productive, functional and obstacles being removed. The other members in the team would ideally have a varied array of skills. For a project such as our own, we would need experience in programming, user interface design and report writing.

CHAPTER 1

My first idea for the project proposal was the idea of creating a nutritional app that would be able to track user's calories, macros, and nutrition. It would also set goals depending on the level of fitness the user may want to achieve. The website would allow users to input information about the foods they eat, as well as their physical activity, and it would provide them with useful information such as the number of calories they have consumed and burned, as well as the nutritional value of the foods they have eaten.

One of the key features of the website would be the ability for users to set specific goals for themselves, such as losing weight or improving their overall health. The website would then provide them with personalized recommendations and support based on their goals and the information they provide. For example, if a user is trying to lose weight, the website might suggest that they eat fewer calories and provide the user with a specific meal plan to achieve these goals and exercise more to hit the users specified goal.

The fitness features would include workout plans and exercises which are also customisable to directly fit the user's specified needs. E.g., if one of the user's goals were trying to lose weight while also trying to be more fit (stop being out of breath) the website would generate a meal plan with a calorie deficit and a workout plan that's cardio focused.

Another important aspect of the website would be the ability for users to track their progress over time. This would be done using charts and graphs that would show how their weight, body fat percentage, and other metrics have changed over time. This would let users check their past and allow users to see how their efforts are paying off and make any necessary adjustments to their habits if their goals aren't reached in order to achieve their goals more effectively.

Overall, my website project proposal aimed to create a tool that would help users improve their fitness and wellbeing by providing them with the information and support they need to make healthy choices. By tracking their nutrition and calories, users would be able to take control of their health and make positive changes that would help them achieve their goals.

The reason my project was rejected was for various reasons

One of the reasons my project proposal was rejected, and we chose to make a football statistics app was because in terms of scalability the football could've been expanded on much more as opposed to the nutrition app e.g., we could have much side features e.g., gambling coaching programs etc.

Another reason was that we were simply more familiar with football as opposed to nutrition and would have to do much more extensive research to familiarise ourselves with the field of nutrition and fitness,

another issue that was raised was that we could not find an open-source fitness-based nutrition app that held the information that was needed.

Our proposed project is a gaming website that will provide reviews, news, and videos related to the latest games that are out. It would be a one-stop destination for all things gaming.

The website will feature reviews of the latest games, written by experienced gamers and industry experts. These reviews will provide in-depth analysis of the game's mechanics, graphics, story, and replay value. In addition, the website will feature news articles and videos on the latest gaming industry trends and events. This will include coverage of major gaming conventions, new game releases, and updates on popular games and franchises. And will also provide a review section where streamers can upload their reviews and rate it on a scale of one through ten and viewers will be able to interact with these reviews in the form of comments or likes, these features would also be included in the critic's reviews

One unique aspect of our website will be a section dedicated to gamer drama and controversy. This will provide a platform for gamers to discuss and debate the latest hot-button issues in the gaming community, such as toxic behaviour, in-game microtransactions, and censorship. I believe that this would foster a sense of community and engagement among our users. It would redirect traffic from platforms like Twitter and YouTube to the website and attract users with the purpose of freely discussing opinions between users where they can freely debate. There would be a moderation feature with certain words blocked out to prevent toxicity on the platform.

In terms of technical details, the website will be built using a popular content management system such as WordPress. This will allow for easy updates and maintenance of the website's content. The website will be optimized for both desktop and mobile devices, to ensure that users can access it easily on any device this would maximise the number of users active on the platform.

Overall, our gaming website will provide a valuable resource for gamers of all levels. With in-depth reviews, engaging news articles, and a community forum for discussing gamer drama, we believe that "Gamer's Guide" will quickly become a go-to destination for gaming enthusiasts. And will be a great platform to showcase high quality games.

The reason my project was rejected was for various reasons

The first reason was that I was the only member experienced with using WordPress this would mean that all the other group members would have had to learn what WordPress is how to use it and

Another reason was that again we were simply more familiar with football as opposed to the gaming industry as there was only me and one other member of the group, this would mean the rest of the group would have to do much more extensive research to familiarise themselves with the gaming industry.

eBook Library (Calibre) - A website with an eBook library that would likely end up directly linked to Z-Library that has since been banned and went under investigation by the FBI.

Sky Scanner App – An app that you could use to take a picture of the sky and based on the location of stars on the sky it would tell you what constellations you are looking at. As an extra for this proposal, we were considering adding the functionality to view real time location of satellites and space stations such as the ISS.

Football Statistics Tracker – A website that would show historical football statistics that with time evolved to a website that would let you search a specific keyword and based on that keyword it would show you football related statistics-based n a player or league or team. The original plan was to have several pages to browse teams, representations, players etc.

The chosen project in the end was the football statistics Tracker

The reason this project was chosen in the end was that we were all familiar with the technologies that would be used developing the web application. We all had experience with web-based application from a previous module and we had also done databases so we knew how we would store the data safely and effectively. The football statistics app also had very high scalability e.g., we could add more complimentary features like having the statistics displayed, coach mode would have all the complicated statistics displayed so the coaches could view the data to improve their teams, a gambling section for registered users to bet on their favourite teams. We also agreed that we could get most of the core features done within the given timeframe. We also had the project more planned where and how we would get the actual football statistic from we concluded that using a web scraper would be the way to go and then storing it in cloud.

Overall, the project was chosen because we had it planned out more than the other proposals. we were more familiar with the technologies used than the other projects and was more realistic than the other proposals as in we would get it done within the timeframe.

And finally, we were all just more familiar with football.

Other proposals from other members were:

eBook Library (Calibre)

A site having an eBook library that would probably wind up being directly linked to Z-Library, which has subsequently been prohibited and is the subject of an FBI investigation.

Sky Scanner App

You could take a picture of the sky with an app, and it would recognize the constellations you are looking at based on the stars' positions. As an additional for this concept, we were thinking of including the ability to track the whereabouts of satellites and space stations like the ISS in real time.

Football Statistics Tracker

A website that would show historical football statistics that with time evolved to a website that would let you search a specific keyword and based on that keyword it would show you football related statistics-based n a player or league or team. The original plan was to have several pages to browse teams, representations, players etc.

CHAPTER 2 (1500 words)

Ideas and Goals

At our first meeting, we agreed to have a brainstorm of ideas. Everyone got a chance to talk about their idea for an app, from which we all made a decision which project to pursue. While the project developed from the football statistics idea which Jamie brought to the group, the application we decided to create differed substantially due to our initial group brainstorm. Jamie's idea focused more on the historical statistics element of football, where our solutions would act as a storage and display of information in a user-friendly way, from various data sources. What we came up with as a group was to focus on the live statistics part rather than the historical statistics and records side. We decided that live statistics would be a more useful idea and perhaps a feature of historical matches can be added into our app along the way.

To implement live statistics, we needed a source to supply us with data, storing this data and displaying the data to the user on request. This would be the primary functionality of the website. There was a secondary idea to add a feature to the application which allowed users to create a team on the site and allowed a manager or a team statistician to add statistics about a match to the site for players on a team to view statistics about themselves on the same platform as they can view statistics about their footballing idols/peers. This combination of localised statistics and global football stats gives users extra reason to use the application. With Scrum, different aspects of the ideal project are prioritised in different stages of the course. The ability to create a team and upload statistics to the players was placed as a low priority which we would add only if the live, global statistics aspect of the project was at a highly functional level.

The idea of a live football statistics platform already exists online, however we realised there were clear flaws and disadvantages to each of the existing sites. We believed we could create something which harnessed the positive aspects of these existing sites and combined them to create something even better. Due to sports betting and changes in how the modern fan operates, the live sports statistics has emerged a huge market. Our group understood that different users of the application are looking for different things. Some users do not understand the intricate statistics of football whereas others have in-depth knowledge of football statistics and are using the application to get detailed information. Therefore, a crucial factor of a good application should not confuse a less informed user, but the detailed statistics should be also easily accessible to those who desire them.

We understood that we would find it difficult to give statistics on as many leagues as existing applications such as LiveScore, due to their large monetary backing. We also were aware that the Premier League website was incredibly effective in terms of delivery quality. The site is easy to use and looks visually appealing. However, it only gives information about one league (20 teams). Therefore, we knew our selling point needed to be about delivering a reasonable quantity of teams and leagues but in a very appealing manner. Our solution ideas combine some features of the best performing websites out there. A one-for-all search bar would simplify the user's interaction with the webpage, cutting down on browsing time. Easy to access dropdown menus, adjusting to multi-device resolutions. Visually pleasing graphics, presenting data in a non-complicated, legible manner.

The image shows two main sections of a sports statistics website. On the left, there's a sidebar with various competition logos and names: World Cup 2022, England, Spain, Italy, Germany, France, Netherlands, Scotland, Champions League, Europa League, Europa Conference, World Cup Qualifiers, Euro 2024 Qualifiers, International, UEFA Nations League, UEFA Women's Championship, Women's World Cup, and Women's Euro. Below these are two live match cards: 'Group E World Cup 2022' (Germany vs Japan at 13:00) and 'Group F World Cup 2022' (Morocco vs Croatia at 19:00). On the right, there are two detailed player statistics tables. The first table, 'Goals', lists Erling Haaland (18 goals) at the top, followed by Harry Kane (12), Ivan Toney (10), Aleksandar Mitrovic (9), Rodrigo (9), Miguel Almirón (8), Roberto Firmino (7), Phil Foden (7), James Maddison (7), and Leandro Trossard (7). The second table, 'Assists', lists Kevin De Bruyne (9 assists) at the top, followed by Bukayo Saka (6), Gabriel Jesus (5), Alex Iwobi (5), Dejan Kulusevski (5), Bernardo Silva (5), Christian Eriksen (4), Andreas Pereira (4), James Maddison (4), and Ivan Perisic (4). Both tables have 'View Full List' links at the bottom.

LiveScore – Live Matches Page

Premier League – Player Statistics Page

Back End

Sourcing the data for our website was our primary concern, as that was the main functionality. We needed a reliable source which regularly updates information. Initially we were certain would use an API (Application Programming Interface) to produce our live scores and league tables. Using an API means we can access large databases with current information without anyone on our team manually making alterations to keep the website updated. These APIs are supplied by companies who gather data, and in most cases, sell it to users. There are multiple APIs available online which we could use to give us score and table updates, the difficulty was finding one which was free to use, as our project does not have any financial backing. Another issue was finding an API with a database that matched our expectations for quality and quantity of information supplied. We knew we could take the easy route and make our application limited to the API size, but we decided that was not good enough.

While research was being done on APIs, Bartek learned about an alternative for sourcing data through Web Scraping. This was not familiar to anyone in the group, and we decided to do further research into how this worked. Our research showed us that web scraping is the act of extracting information from a website and exported to be used elsewhere. Using this process, we could harvest information off a website which is already updated by an API, rather than needing an API of our own. The web scraping could be automated, therefore it would appear no different to the user than if we had our own API. As our API uses data, which is available on another website, the question of what makes us different may be asked. Again, we hope we stand out from this competition based on our tidy delivery of the information, legible to anyone.

```

urlLeague = "england";
const url = 'https://www.samewebsite.com/league' + urlLeague;

//Scraping Premiere League Table
tableLog = [];
(async function scraper(filename, options) {
  const response = await axios(url);
  // connection status check
  // response.status === 200 ? console.log("Table connection: "+response.status + " (OK)") :
  console.log("Table connection: "+response.status + " (ERROR)");
  const html = await response.data;
  const $ = cheerio.load(html);
  const allRows = $('table#bttable tbody tr');

  const replace = (s) => {
    const chars = ["\t", "\n", "\r", " "];
    for(let c of chars)
      s = s.replaceAll(c, "");
    return s;
  }

  allRows.each((i, row) => {
    if(i > 0) {
      let data = [];
      for (let j = 0; j < 10; j++)
        //fills array with data from each column
        data[j] = replace($(row).find('td')[j].text());
      if (data[0] === '') {
        tableLog.push({
          "Place": data[0],
          "Team": data[1],
          "Games_Played": data[2],
          "Wins": data[3],
          "Draws": data[4],
          "Losses": data[5],
          "Goals_For": data[6],
          "Goals_Against": data[7],
          "Goal_Difference": data[8],
          "Points": data[9]
        });
      }
    }
  });
  const tableName = $('h1').text().replaceAll(" ", "_");
  console.log("Scraping "+tableName+"...");
  const csv = new ObjectsToCsv(tableLog);
  await csv.toDisk("./data/temp/" + tableName + ".csv", options);
})

```

Scraper Code [fig. 2.1]

We implemented the Web Scraper in JavaScript using Axios and Cheerio. Below is a snippet of how a basic version of our code works. Once a table is scraped, we extract all the information and populate a new JSON file to store our information (see fig. 2.1). The JSON has all information necessary to make a detailed table for users to view. Some information is not necessary to be viewed by a regular user but will be useful to us in the future for the “statistics predictions” feature to be implemented in future versions (more on that in chapter 3). Variations of tables can also happen, where a user can request to see a less or more detailed view of a table. This can be useful for more advanced/passionate users who require more information.

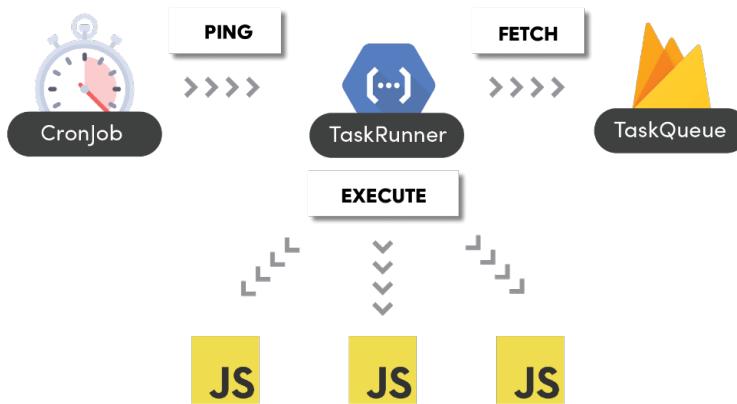
Given all this information, it is a problem to store it on a local drive. Therefore, another technology we used is Firebase Firestore to actually store this information. It works like folders which are called “collections” where entries with data scraped are populated and stored. This information can be later accessed by a couple commands through JavaScript. This feature is crucial for the app as we will pull from the database quite often, every time a user makes a request from the search bar. The data is automatically sorted and given id’s to be easily found or altered in specific feature options. Below is a

picture of the database in the Firebase console. Collections are populated and fields are created in each entry with the data given.

football-webapp-cs353	statistics_data	VCZ3fQlw0JZJ5kQeMGar
+ Start collection	+ Add document	+ Start collection
statistics_data >	VCZ3fQlw0JZJ5kQeMGar >	+ Add field
		Draws: "" Games_Played: "" Goal_Difference: "" Goals_Against: "" Goals_For: "" Losses: "" Place: "" Points: "" Team: "" Wins: ""

Firebase Collection Example [fig. 2.2]

Whilst developing this feature, the World Cup began and we realized the need for automatically updating tables on a timer, so users have live statistics without having to request it. This feature was implemented using a delay and ran on a Firebase hosted server. Wrote in JavaScript the code, is again, similar to our web scraper differing only in a delay timer running on an interval (for this project needs anything more than 24 hours is excessive). See picture below for an explanation.



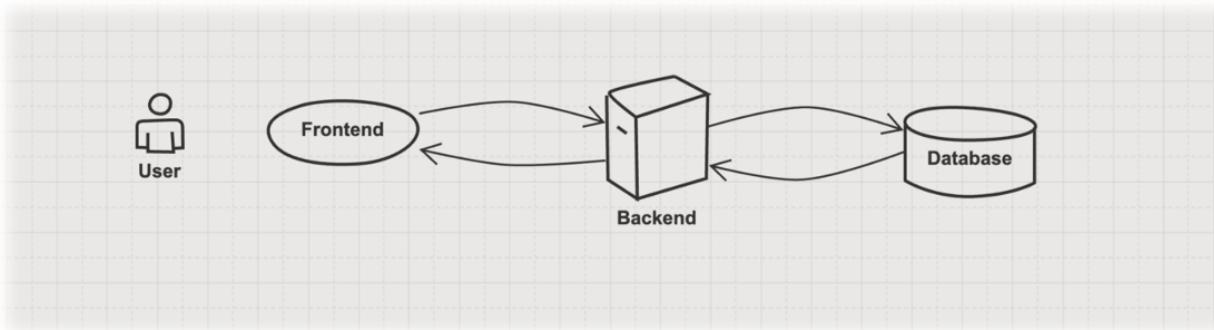
Using Cronjob to Schedule Tasks [fig. 2.3]

Another key feature we implemented involved the onboarding process of the user. We must have a way to let the user register and login on request. On input, this information also is stored in a separate collection

in the Firebase Firestore, where it's easily accessed by a similar method as mentioned earlier (see image #.#). A user can have more than just their registration data in the database. Fields can be populated with favourite clubs, newsletter sign ups alerting the user of changes in table statistics and much more discussed later. Firebase offers authentication services which takes care of security and login functionality.

From having a user registered, we are able to provide more content for the user to enjoy. We can log specific information to make our website more personalized and dynamic. A customized experience of our website makes the user more engaged and lets the application be used to its full potential. Future features are explored in chapter 3. Including a user also means we can easily implement a Coach variation. A minor change in permissions lets us include team organization features and much more. This would include all of the "Guest" and "User" permissions, excluding only the "Admin" ones, where the administrator has access to alter more secure and sensitive information such as account information, inner workings and/or console.

These are the basic features we have implemented through the Back End code which are necessary for the application to be of working order. Further features to explore are discussed in chapter 3 of this document. Although the features implemented work perfectly on their own (running as scripts), they mean very little to users. The data we have collected needs to be presentable and legible for a good user experience. This requires everything to be connected and ran in sync through the Front-End part of our website.



Connection of our Code [fig 2.4]

Front End

For brainstorming and the front-end development guidelines, we decided to develop wireframes. We allocated this work to Conor, who in turn presented the following Wireframing:



Welcome Screen [fig. 2.5]



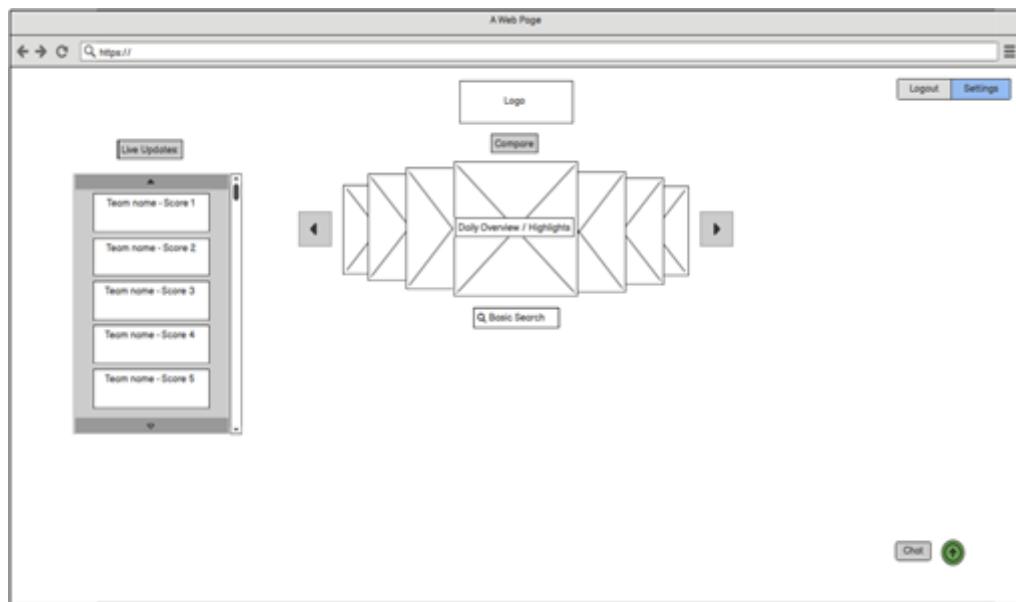
methods of Logging in displayed to user [fig. 2.6]



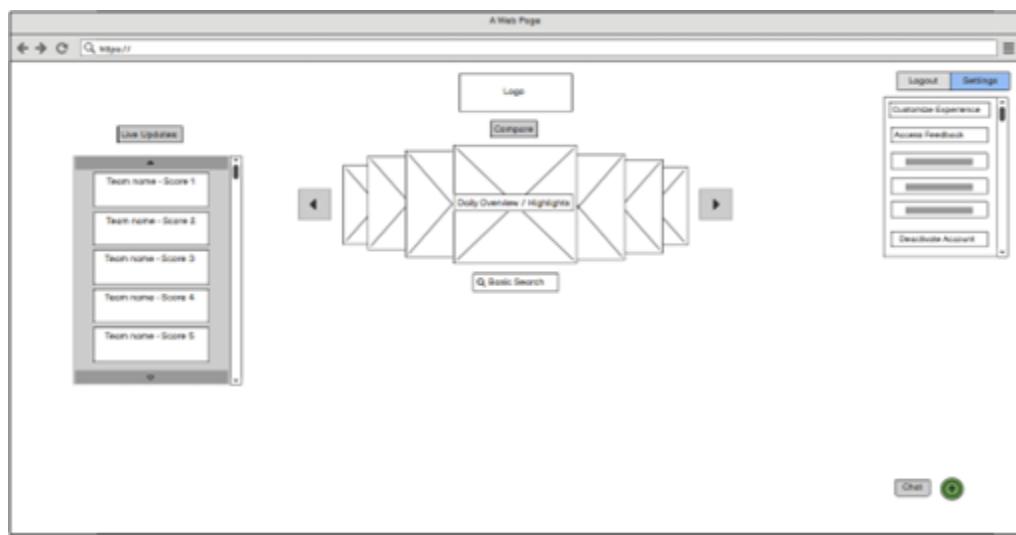
Social/Email Login [fig. 2.7]



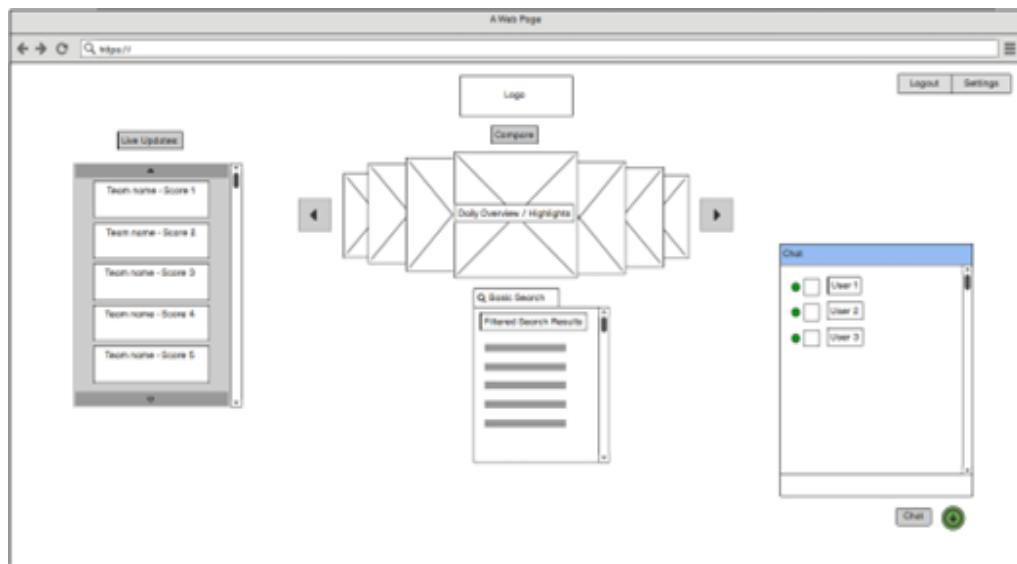
Sign Up using Player/Coach Code [fig. 2.8]



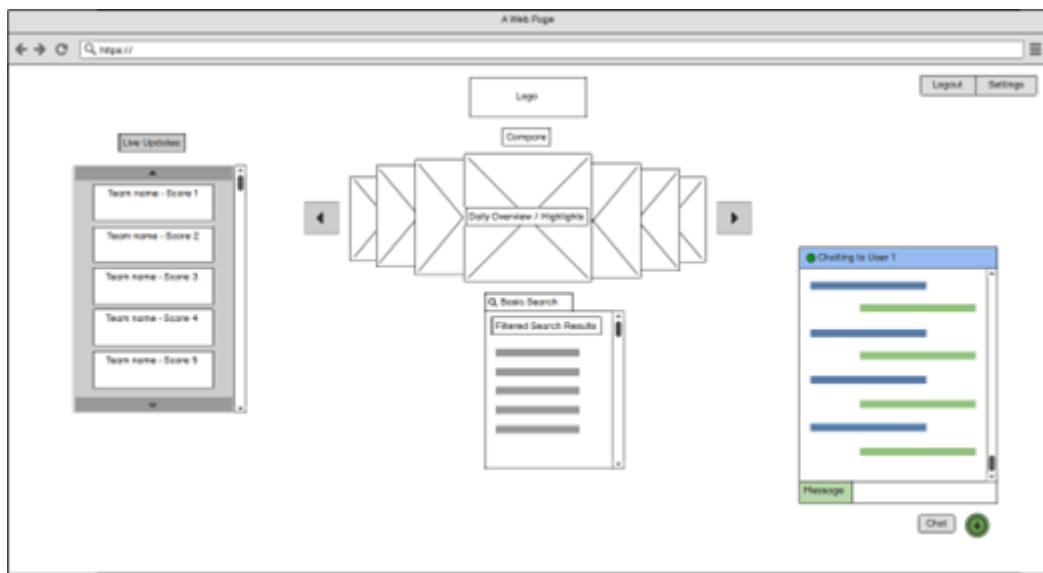
Main Menu [fig. 2.9]



Main Menu 2 (Drop Down Options) [fig. 2.10]

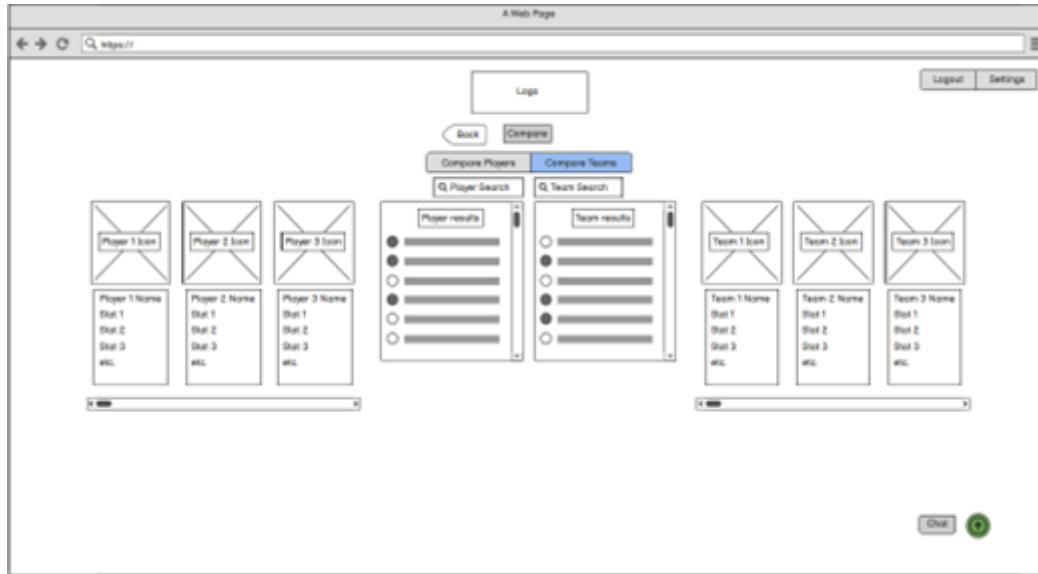


Main Menu 3 (Chat System on Request) [fig. 2.11]



Main Menu 4 (Chat System Conversation) [fig. 2.12]

Main Menu 4 (Chat System Conversation) [fig. 2.12]



Comparison Tool (Compare Player/Team Statistics) [fig. 2.13]

The application used to design this wireframe was Balsamiq. Option prior to this were applications named Adobe Figma and XD but given Balsamiq's simplicity and more professional approach to wireframe design, ultimately this was the application chosen for our web application wireframe design.

This wireframe is an estimation of the design of the application. While the end application design may differ, it will resemble as closely as possible to the designs included above. This wireframe provided our team members with a good estimate of what to expect in the end-product of our application. We recognised that the wireframe, in non-academic environments, is intended primarily for the client; however, in this case it was designed to provide each of our team members with an expected application outcome. We used these as a guideline to furtherly develop our application.

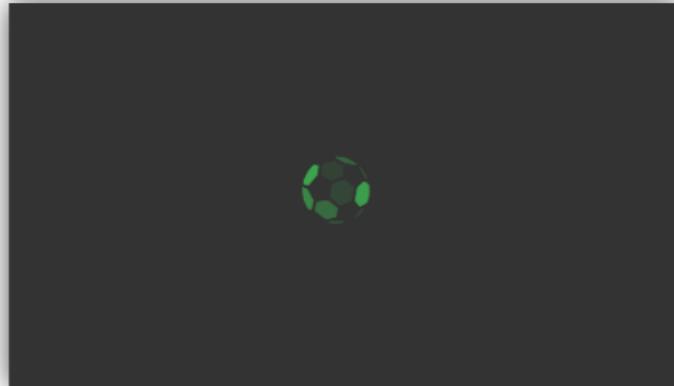
The main technologies we used for the front end of our application are HTML, CSS and JavaScript. Although we had the option of using external libraries such as React or Angular, we used the plain language versions as we were already familiar with these. We believe a solution using these libraries may be more efficient and is a topic to explore in further developments, but for the number of features implemented up to date, our methodology works without problems.

As our development started in Gitlab for collaboration, all of our assets are already uploaded and hosted there. These assets (images, videos, GIFs etc.) can be referenced by a URL in our front end and therefore there is no need to host them again on Firebase.

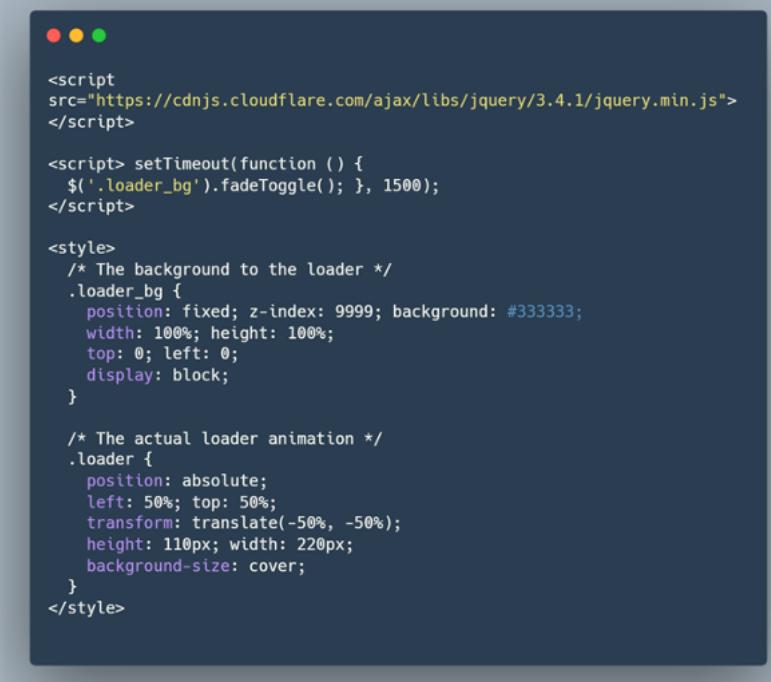
 Upload New File	d421842c	
BARTEK KOWAL authored 1 week ago		
<hr/>		
Name	Last commit	Last update
..		
 LoadingAnimation.gif	Upload New File	1 week ago
 Logo.ico	Upload New File	1 week ago
 Logo.png	Upload New File	1 week ago
 background_video.mp4	Background video compressed.	1 week ago

Assets Uploaded to Gitlab [fig. 2.14]

We have implemented a loading screen to allow images to be loaded in the background before they are presented to the user. We developed the GIF animation in Adobe After Effects using opacity keyframes on an SVG image sourced from Google, paying particular attention to the Usage Rights of images. JavaScript and CSS were crucial for the seamless transition and animation of the loader screen (see image 2.16). After 1.5 seconds, a seamless transition into our website is made and a user is brought to the home page.



Loader Animation [fig. 2.15]



```
<script>
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>

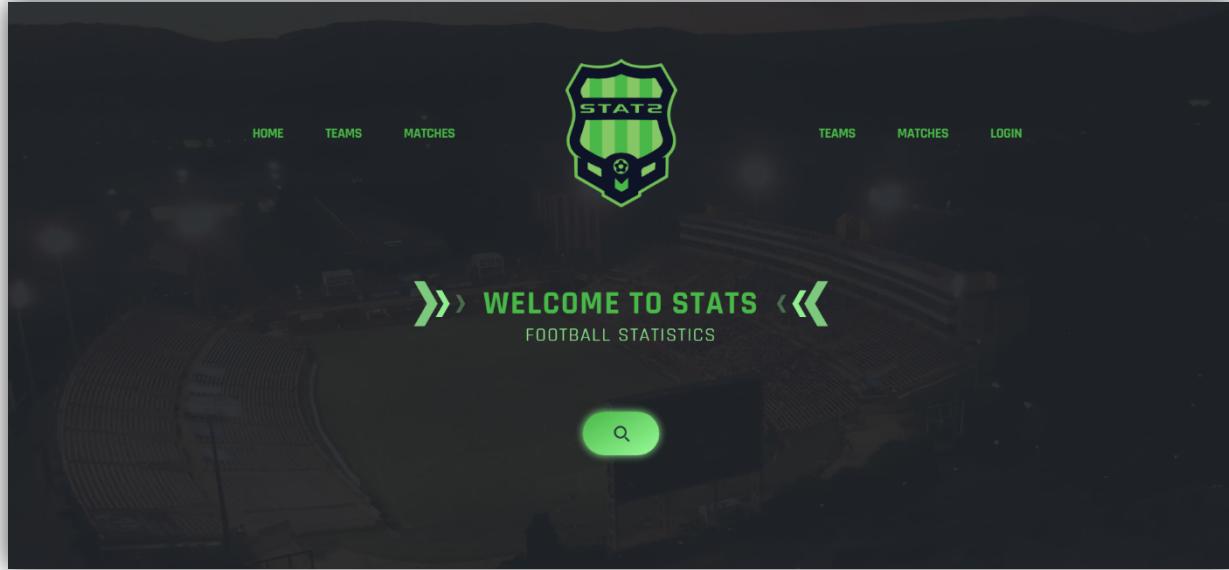
<script> setTimeout(function () {
  $('.loader_bg').fadeToggle(); }, 1500);
</script>

<style>
/* The background to the loader */
.loader_bg {
  position: fixed; z-index: 9999; background: #333333;
  width: 100%; height: 100%;
  top: 0; left: 0;
  display: block;
}

/* The actual loader animation */
.loader {
  position: absolute;
  left: 50%; top: 50%;
  transform: translate(-50%, -50%);
  height: 110px; width: 220px;
  background-size: cover;
}
</style>
```

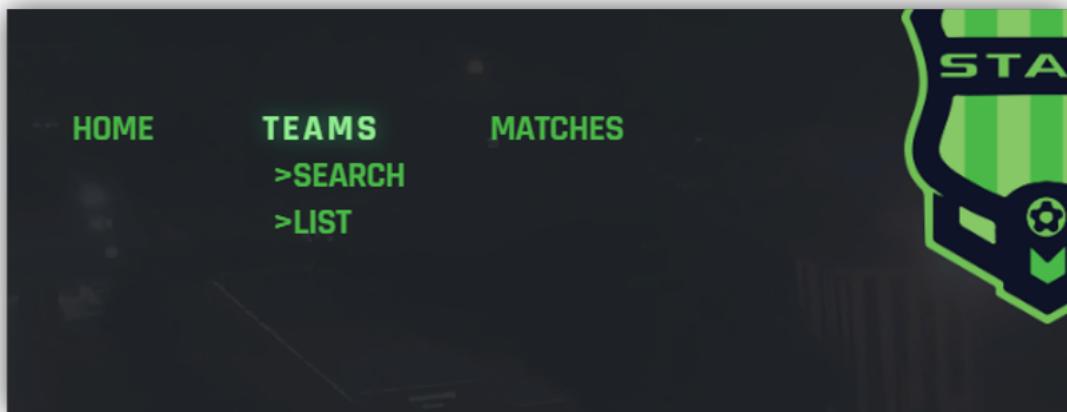
Code behind the loader screen [fig. 2.16]

For our design we chose to take a user friendly, minimalistic style. We worked off of the wireframes to develop a skeleton for our build. Our aim was to make the website as beginner friendly as possible, offering a “one search solution” for all our features. The user can prompt our HTML page to activate a search bar where they type in any query they desire. Whether it’s a team’s statistics or a specific function they want to access, everything is accessible through the automatically predictive, suggestion search bar. The information is sourced from our web scraper and stored in a collection in Firestore as “indexes”. We have hand made our logo using a crest style icon which best suited the theme of “football”. Adobe Illustrator we a particularly useful tool that let us alter any element of our design needs.



Home/Main Page [fig. 2.17]

A connection between the home page and all our other pages is made in the navigation bar through individual drop-down menus/lists to the left and right of our logo. The navigation bar lists are sorted based on the title of each feature. Through CSS, we implemented an “on hover” function which animates and reveals our page links.



[fig. 2.18]

Our page collection includes Home, Table View, Login, Register, User Profile, Error 404, Leagues List, Contact, Our Team. We chose to host our files on Firebase as this proved to be a very fast service which

already integrates great with our back-end developments and connections can be made seamlessly. All of the pages mentioned above can we viewed live on a domain under this address:

<https://cs353footballwebapplication.web.app/>

Testing

On the back-end testing of our application required excessive research of trusted websites to extract data from. With prior permission, we decided on a source and explored tests as to how data can be stored and sorted. Different pages follow different HTML structures. Our main web scraper was tested against various versions of HTML and altered to scrape efficiently. Making sure registration works seamlessly was one of the priorities, we extensively tested registration through outsourcing different people to try and register and log back into an account made on our website. In a similar way we tested the features available which gave us an insight where to improve our UI/UX.

From the frontend, we have tested our website for any wrong redirects and null pointers. We have stripped particular features and decided on a modular one-page solution for the table viewing. A single HTML page now gets populated with dynamic data, which is more efficient without the need of extra redundant pages.

Testing our website speeds online let us make a decision about hosting. We tried multiple services like AWS, GitLab, GitHub and Firebase. From testing speeds and synchronicity with our features, we decided on Firebase. More in-depth testing discussed in chapter 3.

CHAPTER 3 (4500 words)



Miro Screen Capture of Story Map [fig 3.0]

Initial SCRUM meeting (Story Map Creation/Releases/Sprints)

We used Miro (our groups collaboration board can be found under a link [here](#)), as a method of organising our *ideas for* the story map. This was a useful collaboration tool as it allowed us to continue working when we weren't in a team meeting or class. By collaborating, we created the story map above (fig 3.0). It included a list of user features, each prioritized accordingly between the guest, the registered user, the player, and the admin. One benefit of Miro was that we could all be logged on in the same room on separate devices, this allowed us to have a discussion out loud about the priorities while inputting it into a shared document. This saved someone from having to collate 5 individual priority lists into one for the team.

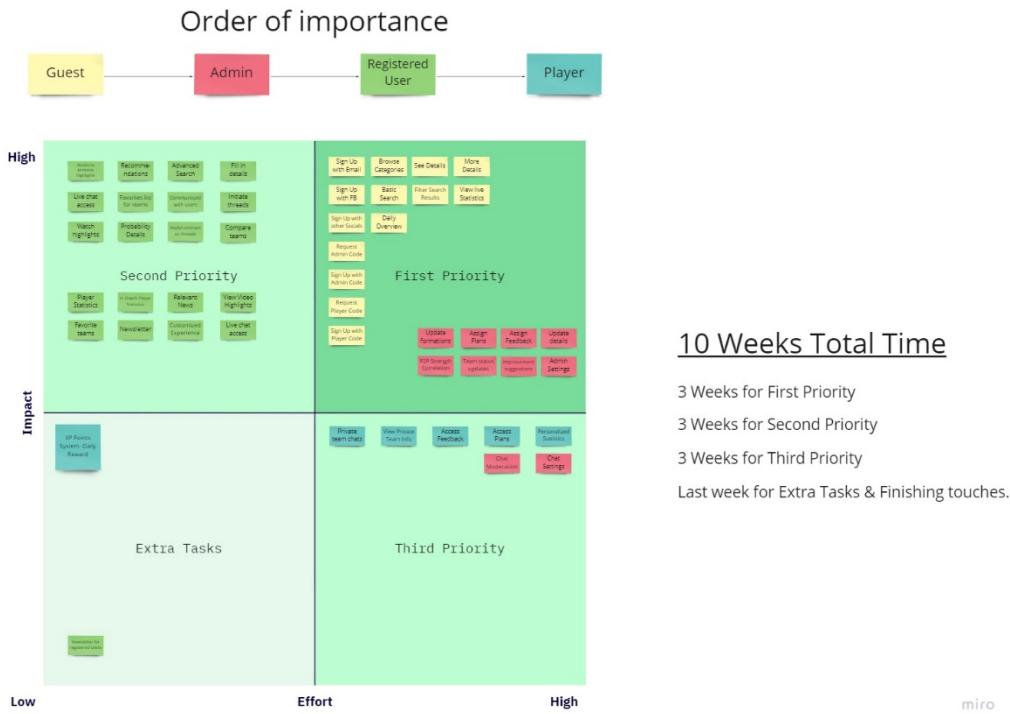
We were aware of the project timeframe and anticipated how certain features might present some issues, so should this become an issue, we were prepared to let go of less important features – at least for the time being. These features could instead be added at a future date. Longevity is important for the lifecycle of an application, so by forgoing certain features for deadline, it allows a stronger focus on the primary application components. This creates a strong foundation for the application, which will benefit the long-term functionality of the app and its future release iterations.

Project estimation – Planning Poker

In the early stages of our project, our intention was to proceed with carrying out a ‘planning poker’ task. This was one part of the project that, collectively as a group, we decided to put our own spin on. We understood the benefits of carrying out planning poker, as it provides uninfluenced estimations and inputs from each of the team members but given that we were in a relatively small team when compared to a real-world team – a team that might consist of many more people – we proceeded in a different direction instead which would benefit us better as a group.

Our substitute for planning poker was Miro. This is a collaboration website that allows us to individually gather our inputs and shape a story map and insights on the project. Given that this is an online website, we were able to interact better with one another, outside the constraints of a physical meet up. By showcasing all our inputs online, we were able to individually revert to the collection of information we had gathered, and if needed, meet up online as a group here to discuss our planning stage of the web application. By opting out of the planning poker and using Miro in its place, we were able to better discuss, prolong the planning stage some more, and if needed, make changes to our plans for our web application. This allowed the team to manage the timeframe of the project more effectively so we could better reach each sprint of our project on time.

Similarly, to planning poker, the benefit to this was to relieve the issue of time pressure on each member of the group. We allocated the ‘difficulty’, or rather the importance of each application component by means of colour coordination (see fig 3.1). Our group collectively agreed that this was a more efficient way of organizing component priorities within our development process. Furthermore, although Miro offered an option for anonymous contribution, the anonymity factor of planning poker was not a necessity for our group project, and so this was another reason as to why we opted not to use planning poker. Despite the omission of anonymity however, our group still maintained an equal consensus as to the direction and timeframe for the project.



10 Weeks Total Time

3 Weeks for First Priority

3 Weeks for Second Priority

3 Weeks for Third Priority

Last week for Extra Tasks & Finishing touches.

miro

Miro Project Plan “Planning Poker” [fig 3.1]

Meeting Minutes

28/09/22 PROJECT SETUP

We had an introductory meeting, got to know one another and discussed various topic of what we wished to do for this project.

Project setup and decision making on software to be used in our development. Discussion of story maps and revision of Moodle material. Brainstorming of features and more research on what is to be included in our project. Setup of GitLab.

05/10/22 SPRINT 1

We used Miro as a method of collaborating within our group as it has proven its benefits previously.

Our group decided on a set of user backlogs and began prioritising tasks by creating a graph with four sections. The sections were divided into first priority, second priority, third priority and extra tasks.

Given this project will run for 10 weeks, we divided it into groups of 3 weeks. The first 3 weeks will be first priority features like user registration and the search feature.

Weeks 4-6 will refine the application by adding second priority features that enhance the user experience. These include access to a live chat, player statistics and advanced searching.

Weeks 7-9 will then allow us to add the smaller, less important features to the application. These features may be less important but again will improve the user experience. Examples of these include a private team chat, chat moderation and chat settings.

In the final week of the project, we will begin putting the finishing touches to the application. These features will come in the form of an XP points daily reward system and a newsletter for registered users.

12/10/22 SPRINT 1

This week, our group began assigning objectives to each of our teammates on the Jira Software app ([x](#)). Some objectives include HTML Setup and Brainstorming, Documentation of Ideas, and research of APIs for Statistic Information.

A wireframe and overall structure will be drafted based off the priority features from last week's user story mapping.

The group account was set up for the Google VM.

We had our first SCRUM meeting with Bartek as the scrum master. Each person discussed their contribution to the previous week's progress and what they intend to do for next week.

Members of the group started some research on API's and other apps/websites with similar information. Alex collated a pros and cons list for some of these apps on the Teams page. We have encountered some challenges with paywalls on APIs.

19/10/22 SPRINT 2

This week, we examined our progress as a group. A wireframe sample was created from our list of features gathered from [Miro](#). As previously noted, in Miro we created a prioritisation of features. From this, it allowed us to construct the beginning stages of the web app design. It consists primarily of search components extracting data from what will be a database of sports statistics/information. It also includes comparison features to compare multiple players to one another, as well as teams to other teams. Additionally, a user communication method was inserted to display a chat system between two users.

For next week, we will get started on getting a search function working so that it can display player and/or team statistics. The search components are a primary component of this web app, so getting a start on this early will be a great benefit to the team project.

Jamie was our Scrum Leader today. He gave a summary of last week's objectives for this week, and we concluded as a group that last week, we previously assigned too many tasks. This has been realised now and we will cut it down a little to allow each of us to better manage our individual tasks for the coming week and sprint.

26/10/22 SPRINT 3

Given how we will not have class due to the midterm break, we scheduled a meeting for Thursday, the 03/11/22 at 11am. This is so we can discuss our progress for the week between each member. Jamie created this meeting on Teams, and we decided on a time during the week off that suited majority of the group.

Finding an API for sports statistics was a challenge for us this week. Conor managed to get a search function working but finding the data to support it is still an ongoing process between the group. Bartek was able to develop some web scraping technology and tested it on numerous websites, but they only worked for a limited amount of time. We have the menu screen of the web app assembled. It has similarities to the menu of the wireframe created by Conor in the early stages of the project.

We had our sprint review this week. We individually discussed our progress and challenges since last week. We talked about our main issue, which was finding a suitable and reliable API. We decided that if we can't find one, we can settle for anything that displays the appropriate data - just as long as it allows

our app to function. In our write up or further presentations, we can discuss our long-term goals for the application which will include a larger scale API. However, this is just a resort, should an API not be available. We discussed this with our demonstrators who understood the issue with finding APIs.

09/11/22 SPRINT 4

This week, we continued working on our project. During the midterm, we met through teams and discussed our project progress. We felt this was necessary as we wanted to maintain our progress as a group. During this week's meeting, we worked together on correcting some coding we had done, including inserting the search function into the project. We had our weekly sprint and each member of our group spoke about their individual progress.

16/11/22 SPRINT 5

We had our sprint meeting where the main discussion involved Firebase, databases, as well as scraping certain data and putting the data into firebase. We also discussed where to host our website and which options are best.

Jamie finished Chapter 0 of the project report and we discussed beginning to piece together chapter 3 of the report. We have some aspects of this chapter done already, so we have to add the remaining parts to it. For example, the burndown chart, testing and future releases.

23/11/22 SPRINT 5

This week we mainly focused on Chapter 3 of the report. There are a few parts remaining that can't be done individually, and therefore have to be postponed until the final stages of the project, like the testing process, the burndown chart, and the final meeting minutes.

During our weekly sprint meeting, Conor was the assigned scrum master. We began by stating the progress of our individual assignments since the prior week. Our report was a primary focus for our discussion. Jamie has been working on Chapter 2 of the report, while Conor has been working through Chapter 3. Given the length of Chapter 3, we have agreed to input different contributions to Chapter 3. This was just to ensure a wider range of discussion within the chapter topics. Alex has been assisting with the report and will be helpful for the progress of each remaining group chapter (2 and 3 – Jamie has chapter 0 complete since the previous week).

Bartek got the scraper working fully, and now we are in the process of starting to link different web pages together. Ishmael has been in charge of additional HTML page development, and he is making good progress on this. He will be working on the log in menu of the application between now and next week. The focus for the next week will be tasks like inserting JSONs into tables, but the primary focus is on front end development and the continuation of the project report write up.

Bartek has researched Firebase. Our project was fully hosted and interlinked with the platform by this meeting.

30/11/22 SPRINT 5

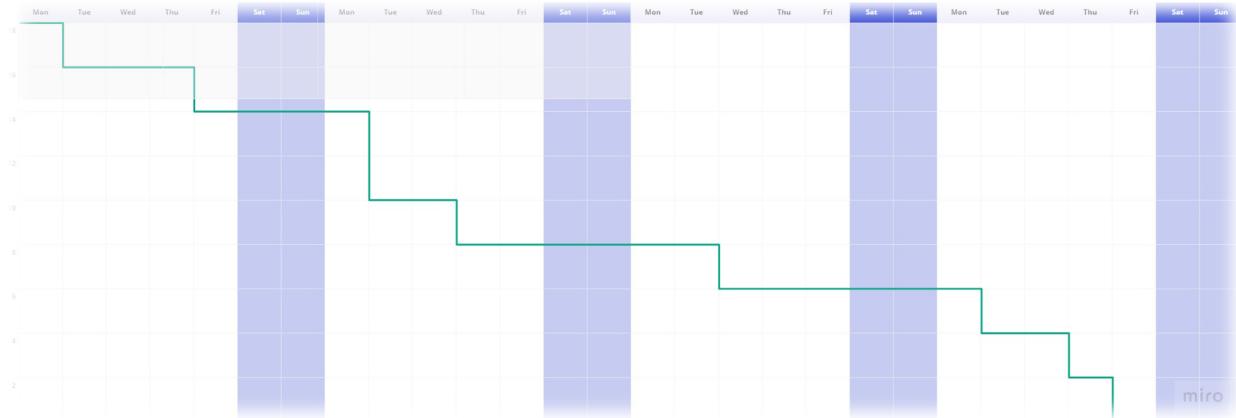
During our sprint this week, one of our primary focuses was the project report. We imported our chapter 0, 2 and 3 progress into a collaborative word document so that each team member would have access to them. We discussed our contributions and our next steps in completing the report. Another focus this week on relation to coding was merging pages together for our project. Furthermore, our team member Mohamed had a part in the Firebase component of our project.

CLEANUP

This week involved a clean-up where each person looked over the work completed and reviewed the tasks most recently completed. This week is a wrap-up week, and our last official meeting before presenting.

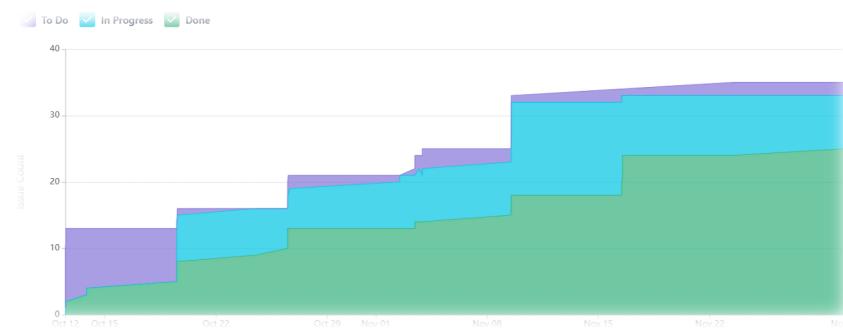
Burndown chart

We used Jira throughout all of our sprints to easily assign work. Each task from each sprint is logged and outputs a formatted graph at the end. We aim for a smooth curve of completed tasks at the end of our project. Our predicted chart for this project is as follows.



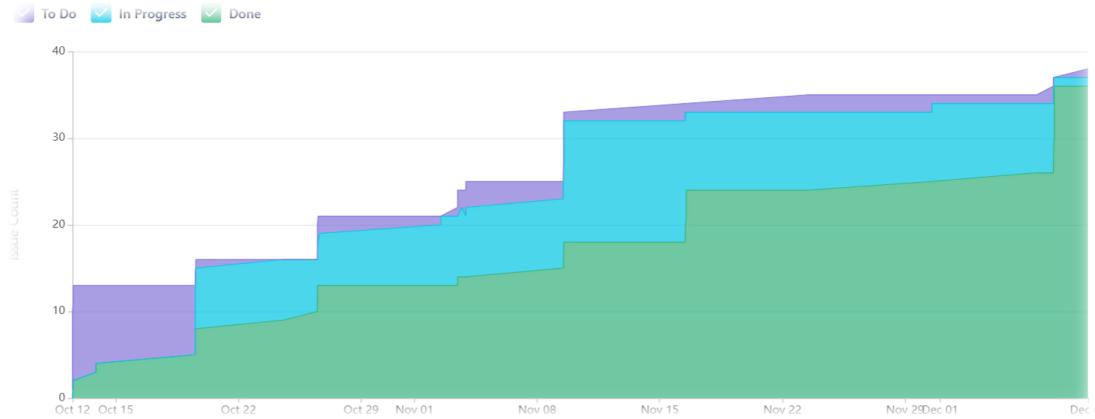
Predicted Graph we Aim for [fig. 3.2]

We hope to have every task completed by the end of Sprint 5. Here is a current (beginning of Sprint 5) graph from Jira of how our tasks developed over time. From this we can observe when exactly we assigned tasks and how many of them were completed. We can see in the chart (fig. 3.3), we can see that as we come to an end of the project, all of our tasks are either “Done” or “In Progress”. This shows steady involvement and completion of work. We are set on a steady course to complete this project by deadline and maybe even before.



Start of Sprint 5 Burndown Chart [fig. 3.3]

The chart below (fig. 3.4), displays our final burndown chart fully completed where green meets our purple graph, meaning all tasks assigned (purple) have been completed (green).



Finished Burndown Chart [fig 3.4]

Testing

We have performed many types of testing during our development process. Our application had many areas that required thorough testing as many things could go wrong in either the back end or front-end parts of the project.

From the back end, we conducted our own testing. We needed to make sure our web scraping code works without any problems. We compiled a list of researched websites that provide statistics relevant to what we need and tried targeting specific tables in the HTML source. Many websites had differing HTML structures where we had to account for different types of tables. These variations all had to be tested against so that we could scrape efficient and accurate data. We ran the web scraper on each one to find out our issues. Similarly, we scraped “indexes” for our data. We had to interlink a user search query to our data entries in order to display the correct information. To do this we created searchable terms with paired id’s which pointed to their matched entries. We scraped a variety of data like team names and leagues, from which a user could search from.

On our front-end side, the experience for the user had to be friendly and very easily understandable. We implemented a “one search solution” that gives the user a predictive search bar that picks from our “indexes” collection. This feature was hard to test for as it’s based on the user’s interaction with our page. Therefore, we had to “recruit” testers for our application. We had a couple people go through our website testing mainly our search function. We observed issues with our UI/UX and updated our website accordingly. The drop-down menus proved to be finicky to play around with in CSS, so we tested a bunch of different options and implemented the best working one.

Upon searching for a term, tables would fill in wrong places. We had step by step debugging, to figure out the error and with the teams help we landed on a solution. Other features explored were login functionalities and registration, where testers tried to register and login with our page. Luckily everything worked perfect

for that. Furthermore, we observed the basic interaction with the pages and changed our HTML and CSS accordingly

Future releases/sprints.

In the early stages of our project, our idea contained many features and components that were intended to be implemented into our end product. However, through each sprint cycle we realized that we needed to prioritize certain features over others. There were several reasons for this – the primary one being time constraints. With this in mind, by having the essential components implemented and working, it would provide the basis to a functioning web application. We also decided that the non-priority features in our web application would not be removed entirely, but merely put on hold for future releases in our web application. Another constraint involved included funding. As we have, we little funding and the website is not bringing any revenue or monetization profit, we had to focus our efforts on the features which were not cost heavy. This involved many “work-arounds” (such as using a web scraper instead of high-cost API). Given that our application is in very early stages, upon completion of main working features we intend to monetize the website through the Google AdSense program allowing us to display non-intrusive advertisements which in return provide a revenue stream linear to our websites traffic. More expansive features could then be implemented.

One of these features included a chat system. The chat feature would be a primary focus for a future release, given its benefits to the web application. It would allow the user to chat with others either individually or by creating a group. One of the reasons this feature was omitted from our web application and pushed back as a future release is because, combined with creating the chat feature, it would also require a profanity filter of some form. We have explored this in the beginning and for a feature like this to work flawlessly, paid libraries such as MobileMonkey is an option for future implementation. Free alternatives offered by HubSpot would also require a lot of coding resources, circling back to our time constraint issue.

Similarly, to the chat feature, there would also be a forum feature that would allow an open discussion between users on the web application. They would be able to create a forum and reply to others. This forum would almost be more in need of a filter, like the chat, as forums are publicised much more than a chat. Currently the best way to do this is by hiring an administrator to overlook the posts into the forum. Omitting this would create a problematic and toxic forum environment, hence why this feature was also pushed back. Perhaps through a future update, both the chat and the forum feature might be added together, given their similarities. A shared administrator or program overlooking the communication features would both fall under our time and funding constraints.

More features in the form of future releases will include smaller, less important features, but by adding them, the user experience will be improved. This will include recommendations based on the user's sporting interests and a vastly more personalized experience. This could be in the form of personalized statistics based off the user interests. A requirement for this feature is essentially data. A big data collection bank would allow us to make comparisons and in turn, suggestions of content to the user.

A statistics prediction feature is one that we really hoped to implement into our project. This is the feature that could set us apart from the competition in the football statistics space. Using algorithms and gathering previous data from matches we could predict scores of upcoming games, giving gamblers or engage users an idea of how a game might go based on the teams/player history. Once again, in order to carry this out, we would need a big data collection bank which is out of our budget. Algorithms and fine-tuning of the mathematics would also be a time intensive which is one of our strict constraints.

Another feature that would further enhance the user experience is the addition of a highlight feature. This could come in two forms; the first highlight feature providing the user with a list of recent game results. It could include other highlights too, like transfers and football club updates. The second form of the highlight feature would provide video highlights. This would be aimed primarily toward goals and other prominent

match highlights like near misses, player injuries, red and yellow cards, and other similar highlights. For storing such high quality content we need a much bigger database which in turn requires a bigger budget.

Code versioning (Git etc...)

In the first sprint, we have setup a [GitLab repository](#) for collaboration. We have shared this repository to everyone in the group and everyone researched how to use git in order to efficiently work on code together without duplicates or overlaps.

The screenshot shows the main interface of a GitLab repository named 'CS353'. At the top, it displays '17 Commits', '1 Branch', '0 Tags', and '43.3 MB Project Storage'. Below the header, there's a navigation bar with dropdowns for 'main' and 'cs353 /', and buttons for 'Find file', 'Web IDE', 'Clone', and 'Download'. A commit history section follows, showing a single commit from 'BARTEK KOWAL' made 1 week ago, titled 'Upload New File'. Below this, there are several buttons for repository management: 'README', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Enable Auto DevOps', 'Add Kubernetes cluster', 'Set up CI/CD', and 'Configure Integrations'. The final section is a table of files with their last commits and update times:

Name	Last commit	Last update
.idea	Added styling.	1 month ago
assets/img	Upload New File	1 week ago
css	Updates.	2 weeks ago
js	Sprint 3 changes in group lab.	3 weeks ago
pages	Updates.	2 weeks ago
README.md	Added styling.	1 month ago
package-lock.json	Sprint 3 changes in group lab.	3 weeks ago
package.json	Sprint 3 changes in group lab.	3 weeks ago

Main page of repository [fig 3.5]

This repository served as our main point of development up to Sprint 5, where we moved over to Firebase for final changes a development on a hosted site. The repository was regularly updated with code, where all research and trial-and-errors are logged with each commit. Our commits can be found [here](#), although the code on GitLab is slightly out of date as our most recent version is in Firebase.

When changes were made to code, we had code reviews in our meetings to catch up with the changes to the application. This ensures everyone is one the same page. One downside to Gitlab is the limitation of hosting your project on a domain. Static websites are no problem to host on Gitlab, but for our use cases this isn't optimal. We discovered that Firebase not only is also a great collaboration tool, but also has a huge number of features alongside dynamic hosting.

14 Nov, 2022 7 commits	
Upload New File BARTEK KOWAL authored 1 week ago	d421842c
Upload New File BARTEK KOWAL authored 1 week ago	3e6ac0e2
Upload New File BARTEK KOWAL authored 1 week ago	919af7f2
Delete logo.png BARTEK KOWAL authored 1 week ago	80de98f5
Delete ball-loader.gif BARTEK KOWAL authored 1 week ago	88a9d4c3
Delete bgvid.mp4 BARTEK KOWAL authored 1 week ago	dc4f09b7
Background video compressed. BARTEK KOWAL authored 1 week ago	f3af279e
16 Nov, 2022 1 commit	
Updates. Bartek Kowal authored 2 weeks ago	1cf72191
09 Nov, 2022 1 commit	
Sprint 3 changes in group lab. Bartek Kowal authored 3 weeks ago	d28250dc
03 Nov, 2022 2 commits	
Better web scraper logic. CSV data collection. Bartek Kowal authored 1 month ago	314d6995
Better web scraper logic. CSV data collection. Bartek Kowal authored 1 month ago	ab5a78c5
26 Oct, 2022 1 commit	
<small>1 commit left for reviewing such data. Don't afraid to merge.</small>	

Git commits [fig. 3.6]

Team Management, Team Communication and Remote Working

Our in-person communication consisted of meeting for 2-3 hours every week. This meant our ‘daily scrum meeting’ was more of a weekly scrum meeting. While we had supervised scrum meetings in this class to discuss the individual work we had completed remotely and what we intended to do over the next week, we also used the time together to have informal discussions about the project. These discussions included helping each other with the tasks we were assigned, updating our product backlog, and choosing who would take on the new tasks we encountered. The group worked very well together in the lab, and I feel we had a positive team mentality and bonded well over the course of the project.

The rules of the project dictated that we used Teams as our method of official communication platform. Everyone in the group had experience of Teams so it was relatively easy to communicate on this platform. One of the benefits of Teams was that it allowed us to easily save files for everyone to access as they pleased. Our remote working was very much individual, and while we did have some informal communication and updates, it was much less than we had hoped. Programs like Miro and Jira made collaboration easier and served us as a form of indirect communication where we could always update/add tasks according to our work.

One clear negative of our group was our communication outside of the lab time. This is understandable due to our many commitments outside of the project, such as other modules, part-time jobs, and interviews for work placements. However, it is unclear if there is any clear reason for this lack of remote communication. Often, we would each make great progress on parts of the project in the intervening period between the lab meetings, but we did not communicate some of this progress with the group until we were in the lab. It seems we did not realise this error until later into the project when these habits had become embedded. Perhaps we should have thrashed out this issue in earlier Sprint Retrospective, but it appears to have crept under the radar until later into the retrospective process.

As we were only in the lab together for two hours every week, remote working was crucial. Most of the work completed on the project was completed outside of the meetings and away from the lab. We can never

be sure of how this project would have gone with a superior level of remote communication, but it is possible that more communication online would have allowed us to achieve some of our goals faster.

One very positive aspect of our remote communication was our online meeting during the Hallowe'en Study week. As we were missing a week in the lab, Jamie organised a Microsoft Teams meeting on the Thursday of the week off. This would be crucial because of the aforementioned lack of remote communication. The group made a good effort to be present for the call and the attendees, similar to in our usual lab Scrum meetings, shared what they had achieved over the previous week and their hopes for the period until our next lab meeting.

CHAPTER 4

During my involvement in this project, I was primarily focused on the backend aspects, specifically the database side of things. My responsibilities included working on the design and implementation of the database, as well as ensuring that it was properly configured and optimized for efficient data storage and retrieval.

In addition to my work on the backend, I also assisted with researching the user demographic for football statistics apps. This involved conducting market research to gather information about the age, gender, and interests of potential users, as well as analysing data from competitor apps to identify trends and patterns. The insights we gathered from this research helped inform our decisions about how to design and market the app to appeal to our target audience.

In addition to my other responsibilities, I also helped with the group report's production. I concentrated on improving the report's substance and boosting its word count in order to contribute to the project. I made an effort to make sure the report was thoroughly researched and gave our readers useful information and perspectives. I also paid great attention to the report's arrangement and structure to make sure it was simple to understand and follow. Overall, I was committed to assisting the team in producing a superb report that was both educational and well-written.

As part of my research into the user demographic for Sky Sports, I conducted a detailed analysis of the age and gender distribution of our audience. I found that the largest group of visitors to our website falls within the age range of 25-34 years old, which is an important consideration for our marketing and content strategies. This information tells us that we won't need to worry about simplifying our statistics or language in order to be understood by a younger or older audience, which is a valuable advantage.

Additionally, our research showed that the Sky Sports audience is predominantly male, with 75% of users being male and 25% being female. This data can be useful in shaping our target audience and potentially skewing our marketing efforts towards males. For instance, we might consider using more masculine

colours like blue, black, red, or orange in our marketing materials and slightly adjusting the language we use to appeal more to male users.

In terms of competition, we identified Premierleague.com and Skysports.com as the main rival websites. It will be essential to keep these websites in mind as we plan our marketing efforts and try to differentiate ourselves from the competition in order to attract users to our site. This may involve considering the strategies and tactics that these rival websites are using to target their audience and finding ways to stand out and offer something unique or valuable to our own audience.

Storing data on Firebase has several advantages over continuously scraping data and storing it locally. One of the main benefits is that Firebase provides a cloud-based database solution, which means that the data can be accessed from anywhere with an internet connection. This is in contrast to storing the data locally, which requires the data to be physically present on the device where it will be accessed.

Another advantage of using Firebase is that it offers automatic data synchronization across all connected devices. This means that whenever the data is updated on one device, it will be automatically updated on all other devices that are connected to the same Firebase project. This can be very useful in situations where multiple users need to access and update the same data.

Additionally, Firebase provides a number of security features that can help protect the data from unauthorized access. This includes the ability to set up user authentication and access controls, as well as support for data encryption. This can help ensure that only authorized users are able to access and modify the data.

After considering these factors, I ultimately decided that using Firebase to store our data was the better option for our project. The ability to access the data from anywhere and the automatic synchronization across devices were particularly appealing, as they would allow our team to collaborate more easily and efficiently. The security features offered by Firebase were also a major factor in our decision, as they would help protect the data from unauthorized access.

To import JSON data into Firestore using Firebase, I followed a series of steps to ensure a smooth and successful process. First, I created a new Firebase project and enabled the Firestore service, which allowed me to connect to the database and begin importing data.

Next, I utilized the Firebase CLI (command-line interface) to create a new Firestore database instance. This was an essential step in the process as it provided me with the necessary tools and resources to import the JSON data into the database. To actually import the data, I wrote a JavaScript class that utilized the import command provided by the Firebase CLI. This command required me to specify the path to the JSON file as well as the collection within the database where the data should be imported. The import process itself was relatively straightforward and only took a few minutes to complete, thanks to the intuitive and user-friendly nature of the Firebase platform.

Once the data was imported, I was able to access it within the Firestore database using the Firebase SDK (software development kit). This gave me the ability to retrieve and manipulate the data as needed for our project, which was crucial in order to achieve our desired results.

Overall, the decision to utilize Firebase for importing JSON data into Firestore turned out to be an extremely successful choice for our project. Not only did it provide a simple and efficient way to load the data into our database, but the Firebase platform also made it incredibly easy to manage and access the data within our application. The process of importing the data was remarkably smooth and seamless, and we were able to complete the task at an impressive speed thanks to the impressive features and capabilities of Firebase. In the end, we were thoroughly pleased with the results and would consider using Firebase again in the future for similar projects due to its proven effectiveness and efficiency.

CHAPTER 5 (SUMMARY)

As a group, we were excited to get started on our project and eager to begin building our web-application. Our team consisted of Bartek Kowal, Ismael Plangca, Conor Phelan, Jamie Hoban and myself, all of whom I had the pleasure of meeting for the first time. To kick off the project, we each presented our own project proposals to the group, discussing our ideas and getting feedback from the rest of the team. After some discussion and consideration, we ultimately decided to create a football statistics app, which would allow users to track and analyze various data related to their favourite teams and players.

Once we had our project idea in place, we began the process of delegating tasks and setting deadlines for each member of the team. To help us stay organized and on track, we utilized a tool called Jira, which allowed us to assign specific tasks to each team member and track their progress. This was especially useful because it allowed us to see who was working on what, as well as whether tasks were complete or still in progress. Additionally, we were able to assign ourselves new tasks once we had completed a

previous one, and also list any collaborations with other team members in case we needed to work together on a specific task. Overall, the use of Jira was a great way for us to stay organized and ensure that our project was on track and moving forward efficiently.

On top of assigning work to each other on Jira we also had scrum meetings weekly which were supervised by the demonstrators. We would reveal what we had done during the past week, our progress on our task and when we hope to complete them by if they were not completed, Bartek was doing the frontend and backend I did some research in the beginning but moved to the backend side of the project later on and worked on the database, Alex worked on the testing side of things and also and also did a lot of the research, Jamie also did some research but mainly worked on the report, Conor helped with the report and did some front end, Ismael also did some frontend, we continued this process for the next couple of weeks until we had a finished product. At the end of the semester, we had to present the project as a group, created slides for this purpose and had a meeting to discuss the parts we would each be presenting. After this we recorded a screencast which we all had a part in and Bartek put it together

Overall I think we drastically over estimated what we could do within the time frame, even though we got most if not all our core functionalities done but most of our extra functionalities were not able to be completed, I believe the only way we would've been able to complete all the functionalities we wanted to complete if we hadn't had other modules as just finding the time to do project work was difficult. However, I believe it was excellent experience for me as a developer and I improved my communication skills to the point where I believe that if I were put in a similar situation again, I would excel. My programming proficiency and problem-solving skills have also improved drastically to the point where I believe I will take on a similar project on myself as a personal project.

I had high hopes for this experience, and I'm pleased to say that it exceeded all of my expectations. Not only was it enjoyable, but it also presented me with some formidable challenges that helped me grow as a developer. I gained a deeper understanding of frontend development and learned how to effectively set up and integrate Firebase into my JavaScript code. Overall, this experience has been incredibly rewarding and has helped me become a more confident and capable developer. I feel grateful for the opportunity to have gained such valuable knowledge and skills.

Throughout the course of working on this project, I gained a lot of knowledge about utilizing Git in a professional setting. I did a little study on Firebase with the goal of assisting with its deployment as a third-party host.

It was quite beneficial for me to have this opportunity to work on a web application team project. I was able to work together, exchange ideas, and pick up new abilities because of it. In February, as I get ready for my internship, I can't express enough how grateful I am to have the opportunity to work with such a vibrant and brilliant team. This opportunity has been the pinnacle of my academic career, and I have no doubt that it will have a significant influence on how I develop professionally in the future. I found the experience to be really beneficial overall, and I am excited to develop further as a member of this team.

References:

<https://www.similarweb.com/website/skysports.com/#overview>