Find a Player

(Web application)

By

Mandev Seahra

Submitted to

**The University of Roehampton**

In partial fulfilment of the requirements

for the degree of

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

Abstract

Abstract narrative

Signed (apply signature below)

**Declaration**

I hereby certify that this report constitutes my own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of others.

I declare that this report describes the original work that has not been previously presented for the award of any other degree of any other institution.

**Date:** Enter the date here

**Enter your name here**

Acknowledgements

Acknowledgements narrative

In this report, I would like to thank and dedicate this to my family, friends, classmates and professors for supporting me through my final year and working towards this project. It has been a very tough year for me personally with the family issues I had at the start of the year. I know that this would have been a lot harder if it wasn’t for their support with helping me by giving me the best advice they can and calming me whenever necessary.

Table of Contents

[**1.** **Introduction** vii](#_Toc98426634)

[Research Question or Problem that will be Addressed vii](#_Toc98426635)

[Aims vii](#_Toc98426636)

[Objectives vii](#_Toc98426637)

[Legal, Social, Ethical and Professional Considerations viii](#_Toc98426638)

[Background viii](#_Toc98426639)

[Report overview viii](#_Toc98426640)

[**2.** **Literature or Technology Review** x](#_Toc98426641)

[**3.** **Design or Methodology** xii](#_Toc98426642)

[**4.** **Implementation or Results** xviii](#_Toc98426643)

[Evaluation xxvi](#_Toc98426644)

[Related Work xxvi](#_Toc98426645)

[**5.** **Conclusion** xxviii](#_Toc98426646)

[Reflection xxix](#_Toc98426647)

[Future Work xxx](#_Toc98426648)

[**6.** **References** xxxi](#_Toc98426649)

[**7.** **Appendices** xxxii](#_Toc98426650)

# **Introduction**

Find a player is a web application that will be the main scope of this project - it will be demonstrated as a protocol. This is more suited to being a web application as it would be more accessible to users. There are many other ways to go about this but given the time frame to do this and it being a protocol, it will be a good way to explore multiple computer languages.

Describe your project

This project is all about creating an application for people who love playing sports and for people that may want to get a little more involved in sports through this app. Let’s make an example where a group of individuals wanted to play a 5-a-side football game but may only have a total number of 9 that can play. They would need to grab an extra player to make it even and eligible for a 5-a-side game. Therefore, this application would let the group find a player within the area. This works well from both sides as along with helping a group of individuals out, this application can also help an individual out where they may want to be involved more in sports but do not know of anyone that they can play with.

As expected, everyone is at a different level when it comes to playing, therefore this application would give the chance for the individual to give themselves a rating on how good they may think they are at this sport for example, the user wants to find a game to play. The group of individuals could also score the person so that for next time, this would give everyone a better understanding on the level this individual can play at – this is a better way of matching someone for the game.

## Problem that will be Addressed

The problem that is going to be addressed has already been briefly mentioned before. Sports enthusiasts that have a passion for any sports but do not know of anyone to play sports with. This is going to help solve a real time issue for anyone who do not feel like they have anyone they can do sports activities with. Alongside this is can also help a group of people that have a select group but are missing a few individuals to play a full team sport game.

## Aims

1) One key problem this would solve is the fact that a group of individuals will still be able to carry on as planned with the game they intended to happen. A lot of the times, if there is not the correct number of people for a sport that people are trying to organise to play, most of the time the plan will be to cancel the game.

2) There may be someone out there that would love to play a sport but does not have any friends that may share the same interest as them. Therefore, this is a perfect way for the individual to find a game or someone that would like to play a certain sport etc with them – this creates more opportunities for more people, and this would also keep people active.

3) We may live in a country where a certain sport isn’t as common as it may be in another country, for example being able to play basketball in the UK. Therefore, this would expand the different variety of sports someone can play, and which can lead to creating this sports network that can only get bigger where more and more people would continue to grow with the sport and gain a stronger interest.

## Objectives

1. Creating a homepage – letting the stakeholders know what the web application is all about as the title of the page can only tell the audience a small amount.
2. Creating a login page – this application is based on user profiles and determining if they are good enough to play with a selected group, etc, so everyone must have a profile to use this webpage.
3. Database – this is where all the confidential details etc will be stored along with the email and password which will help the users log in.
4. Implementing a working map into a page – the aim for this is to have a feature where all the users, that have signed to this application, selected details placed on the maps, showing where they are from and maybe how far it would take for them to get to the required destination.
5. A project management tool (external) – this is something that is a MUST when building this webpage. Without using this tool, it can lead to a messy workload so for which a project management tool will give me a structural base to work with. Creating a list of all the required objectives and ticking them off when they are completed, is a useful way to complete any sort of project.

## Legal, Social, Ethical and Professional Considerations

The main issue with building this project would be the obvious thing and that is trying to keep the users’ data private from everyone. The only information that should be shown is the key details such as name or player rating for example. A way to prevent this from happening is only trying to ask for the key information that is required to make this application work. Alongside this, another risk that there could be other similar projects that have done something like this. Therefore, the aim is to make sure that this project is differentiated and has its own unique tools to avoid any copyright or plagiarism issues.

## Background

A lot of people have always had a passion for sports, but some may not get the opportunity in playing it as often. Therefore, this gives them the chance to do so and has many different benefits along with this. In a bigger picture, this would come with a lot of health benefits as it would be keeping everyone healthy and active. While being aware that there are other applications that have the same idea as this, the aim is to make sure that this is different by adding a few more features that these other web applications have not yet discovered. For example, it could be something as little as the map feature and having an instant directions feature showing the user how easy it would be to get from point A to point B.

Goals Soccer Centre is a great example of an application where this sort of idea has already been applied to. This is a common place, discovered in multiple parts of the UK, where people can get together and hire a pitch to play football on. Specifically on the mobile phone application, it gives a little insight into how and why this application is very beneficial and how it is playing sports easy. One thing to highlight from the Goals application is that it mentions user experiences and how it is related to people that may want to use this application. The aim is to identify the weakness across all similar projects and how a change can be implemented for this application to stand out and be differentiated.

## Report overview

In this report, the aim is to tackle how to plan, review and implement this project. Starting off with looking at different technologies such as programming languages and database management tools since these will be the two main things to build this project. Alongside this, we will be looking in depth with analysing key aspects of the code used to design the project, along with other design and different methodologies used to help with the project.

# **Technology Review**

One of the main scopes within this project is that this web application will be database driven. The database most suitable for this project would be Firebase [1] which is a backend cloud computing service which is provided by Google. With the whole database side of things, there were a bunch of different database tools that could have been used such as MongoDB or typically MySQL [2]. However, Firebase is a database tool that is easy to manage, set up and even maintain – it also does not require any additional memory since everything will be stored via the cloud. In addition to this, the transition of integrating it with JavaScript is easily shown by the videos provided on the website [3].

Table 1. Comparing different database models

|  |  |  |  |
| --- | --- | --- | --- |
|  | Firebase | MySQL | MongoDB |
| Pros | -Built-in authentication, hosting, and cloud functions  -Easy to integrate with other Google services  -Real-time database with auto-syncing across different devices and platforms | -Flexible relational database management system  -Self-hosted for greater control if required  -Widely used, with more documented resources | -Supports nested data structure  -very flexible in the sense of it being cloud-based or self-hosted meaning greater control  -High performance with the larger the database |
| Cons | -Querying capabilities limited  - Data modelling isn’t as flexible  -Limited control and ownership due to it being a cloud service | -Requires more attention when setting up  -Less performant with the larger the database  -More maintenance as time goes on | -Constructing queries can be very complex  -No built-in authentication or hosting features  -Limited resources available |

Despite these comparisons, Firebase is the best choice for this project mainly for its cloud functions and the real-time database with built-in authentication and hosting. Since the project is not going to be based on queries and more just obtaining information from the database, this tool is the ideal one to use. Alongside this, the auto syncing function is a great feature as any data can be changed at any given time - assuring clients have all the up-to-date data on their devices. Another key thing is that with MySQL, problems may appear depending on the size of the database, therefore since Firebase is a cloud service, there will be no issues with storing data and even so, a fee can amend the space.

Another tool that we will mainly be focusing this project on is the interactive map. This feature is vital as this is where the users will be able to find their destination, as well as the route to get from where they are located, to the destination. The two different map tools that could have easily been used were Leaflet and the Google API maps. Leaflet is a JavaScript library which allows you to easily create an interactive map, mainly for web applications which can also use the Google maps as a tile provider alongside other provider. And on the other hand, Google Maps API is a set of tools and services provided by Google which allows them to integrate maps into their web or mobile application.

Table 2. Comparing pros and cons between Leaflet and Google Maps API

|  |  |  |
| --- | --- | --- |
|  | Leaflet | Google Maps API |
| Pros | -Open source  -Lightweight and fast  -Easy to customise and flexible  -Supports a variety of different map providers | -Easy to use and integrate into existing projects  -A variety of tools and services including geocoding and routing  - A lot of research done as it’s used by the majority users |
| Cons | -Limited with the different things that can be customised  -Less research and documentation  -Certain features will need additional plugins or services | -Limited customization in comparison to Leaflet  -Less responsive than leaflet  -Can be costly due to high traffic since there’s more users using it |

By comparing the two tools, Leaflet would be the more suitable option. The option of this be open-source library means that it can be customised, and the maps can become suitable for the project requirements without being restricted which is what would’ve happened if we chose the Google Maps API. In addition to this, Leaflet has more options when it comes to different map providers which means the feeling of the map can also be suitable to the project requirements. Finally, since Leaflet is a lightweight tool, it will make it faster to use and more customisable despite the fact you may need to add additional plugins. All in all, Leaflet is the most suitable tool since it will be better to customise the features of the maps and align it to the requirements of the project.

# **Design or Methodology**

Figure 1. Entity Relationship Diagram for DatabaseA picture containing text, diagram, plan, technical drawing

Description automatically generated

As you can see in Figure 1, there are a variation of different entity tables within this entity relationship diagram. The scope of this database is to find a way in which each user can have a role within the project. As shown in the top left, the entity table has the Primary Key ‘AdminID’ – this is where the program will require an admin which will be able to create the specific event that will be published. But within this database design, it is shown that the relationship between the admin entity and the event entity is many to many. Anyone with an account should be able to create an event and be able to remove it, if necessary. Alongside this, the entity with the primary key ‘UserID’ allows users to reserve their spot if they see an event. However, within the RSVP entity, there are a few options that will also need to be passed by the admin. For example, some of the fields have included the rating field. If the admin doesn’t feel as if there is match between the rating of the user and the rating that is required within the event, then they would have the option of approving it or not. The reason for the admin to have access to this part of the project is to show the reliability of how accurate the game of the sport can be. Furthermore, within the RSVP entity, there is one-to-many relation since there can be many events, but each event can only have one RSVP per user since that is the focal point of this project.

Figure 2. Display the User Interface for the Home Page  
A screenshot of a computer

Description automatically generated with low confidence

The consistent pattern that will be shown throughout the whole project, as shown in Figure 2, is the button bar that is highlighted in blue on the top. Since this is a web application, we need to make sure from a visual point of view everything remains consistent throughout so that it is gives the users a good experience where they can easily use the product without needing to work out even the simplest of things. Therefore, even keeping things such as the buttons layout at the top alongside a search bar will give the users accessibility to whatever they may require from the project. Alongside from a users experience, from a developer’s point of view, keeping things consistent is much easier when it comes to things like maintenance because then a solution will be manageable for all the different pages at once.

Figure 3. User Interface sketch of wireframeA screenshot of a computer

Description automatically generated with low confidence

The sketch shown in Figure 3, represents the most important pages in which is going to have the most functions and interaction from users. Again, from a design point of view, the intensions are to go for something simple with a few search bars and forms alongside a big side bar that is appearing on the left of this wireframe. These forms are going to allow the user to enter their details, and this is where the relation towards the database will come into place. Once the form is complete, it will create an event which will not only publish this event into the sidebar, but only submit the details into the database.

This database will have a relation with both the connect and maps page; the connect page is where the user will be able to enter their details that will be sent to the database. Whereas, in the Maps page [4], it should do a similar thing in which there will be a sidebar just like in the previous page but alongside the bar, it will publish this event onto the maps using a pointer. This pointer should be clickable so that when the user clicks on the point, the event should appear and allow the user to take actions – whether it’s removing the event if it’s the publisher or if the user want to join the event.

Another page that is going to be implemented is the Login page [5]. This will allow users to create an event from one end, and another user can join the event from when they sign in. The page itself was going to be left out of the project completely, since it was not a feature that was a priority, but to give it a good user experience, a login page seemed very valid, in this case, to implement this despite this just being a prototype. With the log in page, it will give users an option to sign up where the users information will also be saved onto the firebase database.

The next steps will outline the different approach that will be taken – breaking down each page. In each break down, a discussion will be taking place in which tools and languages will be most suitable for this, along with looking at other alternatives too.

1. Home Page:

The home page serves as the entry point and focal point of the application. It aims to provide a visually appealing and user-friendly interface that engages users from the moment they land on the page. The design follows a clean and minimalist approach to ensure a seamless user experience.

The page layout is structured using HTML, allowing for logical organization of content elements. Cascading Style Sheets (CSS) is employed to enhance the visual presentation and styling of the page. Custom CSS styles are applied to create a cohesive and aesthetically pleasing design. The use of appropriate colour schemes, typography, and spacing ensures a visually pleasing and consistent user interface.

The main goal of the home page is to facilitate easy navigation and provide users with a clear understanding of the application's purpose. The navigation bar at the top of the page contains buttons that link to other pages within the application, enabling users to explore different sections. The layout and placement of the navigation elements are designed to be intuitive, ensuring that users can quickly and effortlessly navigate to their desired destinations.

2. Connect Page:

The connect page focuses on enabling users to create events and connect with other users. The design aims to provide a seamless and intuitive experience for event creation, capturing essential event details such as event name, location, date, and time.

The page layout consists of a form where users can input event information. The form fields are carefully designed and organized to ensure clarity and ease of use. JavaScript is utilised to handle form submission and interact with the form elements.

In addition to form handling, the Connect page leverages the power of Firebase, a tool developed by Google, for data storage and retrieval. When the user submits the form, JavaScript captures the entered data and utilises Firebase's Firestore database to store the event information securely. Firebase's Firestore is a NoSQL database that provides real-time synchronization and offline support. It seamlessly integrates with JavaScript applications, allowing for efficient and scalable data storage. The Connect page leverages Firebase's JavaScript SDK to establish a connection with the Firestore database and perform data operations. By utilizing Firebase, the Connect page ensures that event information entered by users is persisted in a reliable and scalable database. This allows for seamless retrieval and management of events throughout the application. Users can create events and have confidence that their data is securely stored and easily accessible. The integration of Firebase with the Connect page involves configuring Firebase project settings and initializing the Firebase SDK in the JavaScript code. This enables the application to communicate with the Firestore database and perform operations such as adding new events, updating existing events, and retrieving event data for display.

Overall, the Connect page demonstrates the use of modern web technologies such as HTML5, CSS3, JavaScript, and Firebase to deliver a robust and user-friendly event creation experience. The seamless integration of form handling and Firebase's Firestore database ensures efficient data storage and retrieval, enhancing the overall functionality and reliability of the application.

3. Maps Page:

The maps page aims to provide a visually appealing and interactive representation of user profiles and events on a map. The design utilizes the Leaflet library, which offers a comprehensive set of tools for creating interactive maps.

The page layout consists of a map container where the Leaflet map is rendered. The map is centred on a specific location and allows users to explore different areas by panning and zooming. The integration of the Leaflet library ensures smooth and responsive map interactions.

To display user profiles and events on the map, JavaScript is employed to iterate over an array of profile and event data. For each profile or event, a marker is created at the corresponding latitude and longitude coordinates. When the user clicks on a marker, a pop-up appears, providing additional information such as the user's name, address, and preferred sports.

The map's interactivity is further enhanced by associating click events with the markers. When a user clicks on a marker, the map pans to the marker's location, ensuring that it remains in focus. Additionally, the pop-up opens to display detailed information about the selected user or event. These features provide a seamless and engaging experience for users to explore and interact with the displayed data.

The maps page incorporates various technologies and techniques to achieve its goals. HTML and CSS are used for structuring and styling the page elements, while JavaScript provides the necessary logic and interactivity. The Leaflet library serves as a powerful tool for map rendering, marker creation, and map-related functionality.

In conclusion, the approach taken in designing these pages involves a combination of HTML, CSS, and JavaScript, along with external libraries and APIs, to create a visually appealing, interactive, and user-friendly web application. The use of clean and minimalist designs, appropriate colour schemes, and intuitive navigation elements ensures a seamless user experience. The integration of Leaflet and geocoding API enhances the functionality of the connect and maps pages, enabling event creation and visualization on a map. The result is a cohesive and well-designed application that fulfils its objectives of providing easy navigation, event creation, and map-based exploration for users.

Alternative Approaches:

Maps Page:

Instead of using Leaflet, another popular option for integrating maps into web applications is the Google Maps JavaScript API. It provides a comprehensive set of features and extensive documentation for map rendering, marker placement, and interactive functionality. The Google Maps API also offers additional services such as geocoding and directions, which could be useful for enhancing the user experience., if the application required more advanced mapping capabilities, a full-fledged Geographic Information System (GIS) library like Mapbox GL JS could have been utilized. Mapbox offers highly customizable maps, extensive styling options, and support for geospatial data analysis.

Connect Page:

While Firebase provides a convenient solution for data storage and real-time synchronization, other backend technologies could have been employed. For instance, a traditional SQL database such as MySQL or PostgreSQL could have been used to store event data. These databases provide powerful querying capabilities and offer strong data consistency and integrity.

Additionally, backend frameworks like Node.js with Express could have been employed to handle form submissions and perform server-side processing. This approach would involve setting up a server, defining routes, and utilizing an ORM (Object-Relational Mapping) library to interact with the database. If the application required more advanced features like user authentication, access control, and real-time updates, a full-fledged backend-as-a-service (BaaS) platform like Firebase or AWS Amplify could have been employed. These platforms offer not only data storage but also user management, authentication, and additional services like push notifications.

Conclusion:

In conclusion, while the chosen approaches of using Leaflet and Firebase for the Maps and Connect pages respectively are suitable for their intended purposes, alternative methods and tools could have been employed to achieve similar or enhanced functionality. The choice of tools and technologies ultimately depends on various factors such as the project requirements, scalability needs, available resources, and personal familiarity. Exploring alternative approaches provides valuable insights into different technologies and allows for making informed decisions based on the specific needs of the project. By considering alternative methods, developers can expand their knowledge and skill set, enabling them to select the most appropriate tools for future projects. Regardless of the chosen approach, the primary goal remains the same: creating a compelling and user-friendly web application that fulfils its intended purpose effectively and efficiently.

In addition to the design, the actual implementation of the project will be done in Visual Studio Code. This is a software that has many different resources since many developers tend to use it regularly. This text editor has a variety of different features including extensions that can be downloaded whenever required and the fact that it has a wide range of users that use this software, support would be much easier to get from other developers. The other benefits of using Visual Studio Code is that the GitHub repository can be cloned through this software which would give easy access to push and pull files, along with data from and to the GitHub repository, making the flow of working on this project and sharing this with other supervisors much easier and not to mention that it has a debugger which is very useful when it comes to typing large pieces of code.

As mentioned previously, this project will be using Google Firebase for the database but alongside this, the programming languages the intensions are to use JavaScript, HTML and CSS. HTML and CSS are the most obvious languages to use when it comes to creating the structure, adding content in the pages and designing the layout of the page so this is why these two languages were best fit. JavaScript is a language that can create interactive effects on webpages along with the interpreted language being used for both the client-side and server-side. There are other languages like JavaScript such as Django which is a high-level Python web framework which is also as effective since it also provides a set of tools and libraries from building web applications, but again is a language that is more common than anything else so in terms of the resources. All in all, these three languages work well together and is usually what developers tend to use when coming to creating a web application.

In order to keep record of all the steps taken with this project, the project management tool that will be used is MeisterTask to break down this process into steps so time can be managed wisely with this project. Alongside this, adding different steps on a excel sheet which is being published on GitHub for supervisors to see if required is a good additional way to check different objectives off throughout this project. Again, other project management tool such as Monday.com could have also been used but first, this wasn’t free whereas MeisterTask was, and it had all the tools needed such as creating timescales and being able to check them off when the task was completed – this was more than enough as to having this project management tool.

To conclude, this approach is something that is manageable within the timescale that has been given. Since this project is only a protocol, focusing on making the front-end design to the potential should not be the main priority. The most important thing should be to make the main function working, which is the database communicating between the connect and maps page – creating the event and then making sure that the event is placed on the map within a pointer is the main scope, alongside allowing the users to join the event and notifying the admin of the event. If there is more time than initially anticipated, enhancing the look of the web application more so it can have that user friendly feel to it can be an option to look in to.

# **Implementation**

The first steps that are taken was on how to get a hold of running the program on a server, locally. In order to do this, installing a http server would be required and can easily be done through ‘npm’ which is a package manager from JavaScript applications – it is very useful when it comes to installing libraries and so forth. In this instance, we would need to do the following which will be shown in the figure below:

A screen shot of a computer program

Description automatically generated with low confidence  
[Figure 4]

As shown in Figure 4, the packages such as the http-server was successfully installed. But then in order to start and initiate this package, a command was added into the ‘package.json’ within the ‘script’ section which would allow the server to start. The command that was written was the following:

A picture containing text, screenshot, font, line

Description automatically generated  
[Figure 5]

The command that was written was in line 8, shown in Figure 5. Firstly, it is defining the script called “start” which is then followed by the command “npx http-server -a localhost -p 8000 -c-1”. This line is basically launching the HTTP server using the ‘http-server’ package that was previously initiated. The localhost specifies that the server should bind to the localhost interface, along with ‘p 8000’ is the port number in which the server should listen to. And finally, the c-1 command ensures that the server is always providing the latest version of the file.

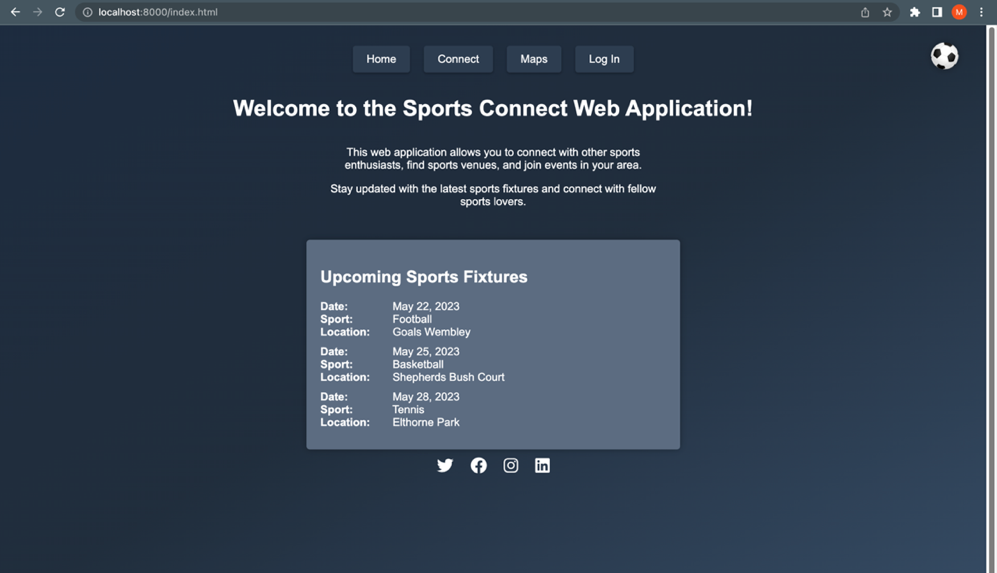
In the figure below is a simple terminal screenshot of me starting the server:

A screen shot of a computer program

Description automatically generated with low confidence **[Figure 6]**

As you can see, the way to initialise the server is basically typing ‘npm start’ in the terminal on visual studio code and that will show the available URL link at the bottom which will allow us to either click on it or go onto an explorer URL search bar and type the following down. And with these easy steps, the project has successfully started on a server. The main reason as to why this application must be written on a server because the database tool, firebase, is something that is being run online since it is a cloud-based database and therefore, if it was being ran locally without a server or without the correct ports, no packaging will be either sent or retrieved to/from the firebase database.

Previously mentioned before, the programming languages HTML and CSS will be used for the front-end design. The main design used for the majority of this project is displayed in Figure 7:

[Figure 7]  
****

A brief explanation of this design was to keep it simple. Four buttons have been created with a simple effect where it allows you to hover over the button you choose to and as you do it, a white line will appear at the bottom of the button– this animation was done in CSS using the hover tool. The idea was to go for a theme that was appealing in the sense of easier for the users to see and interact with; not showing too many colours as at times it could confuse and be too much for the user.

Initialising the Firebase Database

The initialisation of the Firebase database inside of the database can be found on the Firebase website. It consists of a variety of imports that can be added throughout the implementation stage. On the website, once the project has been created, it will give the users a pre-made configuration object that will just need to be copy and pasted inside of the JavaScript code and this will be called using the ‘firebase.initializeApp()’ method. Once this is done, you will be ready to use Firebase services such as Authentication, Realtime Database etc just by referencing the ‘firebase’ global object.

Starting up the Web Application

To start off with, the user cannot use any of the pages unless they have signed in. This was done so that the users who do create an event, can keep their privacy from the public eye and keep it strictly for the other users within the application – it can also be beneficial in terms of the user’s safety and wellbeing. The figure below will show the JavaScript function called ‘signUp()’ which is constructed for the user to create an account to access the web application.

[Figure 8]  
A picture containing text, screenshot, software, multimedia software

Description automatically generated

This function employs asynchronous programming techniques to manage operations that involve waiting for responses from external resources which is shown in line 114, with ‘async’ being the word indicating this. Within the function, two variables have been declared which will have the values that have been inputted by the user in the HTML form – this information will be gathered by using ‘document.querySelector()’, and stored in the assigned variables.

The ’try’ block is used to encapsulate the execution of asynchronous operations that potentially could throw errors. The function ‘createUserWithEmailAndPassword()’ is called and has the parameters ‘auth’ which is the imported authentication service, and the two variables that are inputted by the user in the sign up form. This function helps create a new user with the provided details and if there are any errors which occur during this process, the ‘catch’ will block this error which will also be logged into the console during this process. If there is no error, then the data will be successfully added to the database.

Alongside this, there is another function called ‘updateProfile()’ which passes the parameters of the display name (user input inside the form) and the ‘auth.currentUser’ object. With this function, it updates the user’s profile information and just like with the other function, it will log any errors that this part of the application may face. If all of this is successful, then the user will be redirected to the home page, meaning that the application is ready for the user to use.

For reoccurring users that have already signed up, they will just need to direct themselves to the index page. On this page, it will contain the login form which was created in HTML. The reason this is chosen as the index page for users is because of keeping other user’s information private and their safety as this can be shown as a hazard especially when time and location is being shown on the event profile. In the figure below, it will show the JavaScript function logIn() which is for the login form.

Figure 9. JavaScript function for the log in page  


This function follows the same principles as the ‘signUp()’ function where it declares to from a user input, but this time the data will be recorded from the login form. The signInWithEmailAndPassword() function contains an accessToken which is imported from the Firebase library - this is assigned to the user credentials. When the user types in their credentials and if it meets the requirements, then it will have access to all functions within the application. This would be successful within the function when it is called and is passed through the ‘auth’ object which is also a part of the Firebase Authentication – it will scan through this section on the database and see if the credentials are there (example of test data shown in Figure 10). At first, when you enter the index page (log in form), all the buttons displayed at the top, other than the log in button, will be disabled. This disabled method is displayed in the HTML script which then will be removed, as declared in this JavaScript function when the requirements have been met. A catch is also used in this function and will also display the error message or code, if the credentials are wrong.

Figure 10. Displaying the users in Firebase Database A screenshot of a computer

Description automatically generated with medium confidence

In Figure 11, it shows the input of two post codes in the respective text box and once the correct formatting on post codes have been entered, there will be a chance for the user to press the ‘calculate route’ button. Once this is done, there will then appear two pointers showing the start point and end destination, followed by a route in the colour red and the directions will also be declared on the right of the screen. Alongside this, a sidebar has been implemented on the left of the application where all the events are being displayed. These events are created in the connect page which then gets sent to the database and then read and displayed onto the maps page. Each event is inside of their own container – these events are clickable and once the user clicks the event, they will be able to see a change where the location will be published onto the ‘End Postcode’ text box. This will save them time as they would not need to type the location themselves.

Figure 11. Displaying a route in the maps pageA screenshot of a map

Description automatically generated

Figure 12. JavaScript function for the maps pageA screen shot of a computer code

Description automatically generated with low confidence

Figure 12 shows the JavaScript function named ‘maps()’. This function takes the steps into how the program will read the information from the Firebase database and be placed on the sidebar which was implemented in the HTML script. Overall, this function involves user authentication and database operations. The function itself starts off with an ‘if statement’ which at the beginning, tells the program to obtain the ‘getDatabase()’ object which is referring to the Firebase Realtime Database data (shown in Figure 16). Once this is obtained, the program will make a database query using the ‘get’ function and retrieve all the data that is under the ‘/user/ path – if the data is return this will be classed as a snapshot object, whereby if the object exists, the code proceeds to iterate over the values of the snapshot using the “Object.values” function.

For each value, representing a user’s event, the code retrieves the sidebar element using the querySelector methods and appends HTML markup to it containing all the different attributes and information entered by the user in the form when creating the event. Furthermore, there is an ‘onclick’ attribute appended in the HTML markup which will trigger the ‘handleEventClick()’ function and essentially place the location from the object, into the ‘End post code’ text box which can be seen in Figure 11, in the sidebar. If there are any errors that occur during the database operations, a message will be logged into the console. The way in which the routing of the two locations work will be shown in the JavaScript function ‘caluclateRoute()’ which is displayed in Figure 13.

Figure 13  
A picture containing text, screenshot, font

Description automatically generated

Figure 14. JavaScript code for routing two locations  
A screen shot of a computer program

Description automatically generated with low confidence

The calculateRoute() function starts off with getting user input using ‘document.getElementById()’ which are then stored into new variables. To start off with, it utilizes the Leaflet’s Geocoder by which the start and end values from the user’s input will be converted into geographic coordinates. Once this is completed, the function retrieves the centre coordinates of the start and end locations from the result and then it returns the value by creating a routing control – this will outline the route using the coordinates that would’ve been converted. After adding the route to the map, the function listens for the ‘routesfound’ event, which allows it to log and collect the route that’s been calculates – provides useful information to analyse and debug. This is then followed by a marker which calculates the route and makes sure that the marker is matching the coordinates in each position so that the route is displayed correctly for the user.

Figure 15. JavaScript function for the connect pageA screen shot of a computer program

Description automatically generated with low confidence

In Figure 15, it shows the writeUserData() function which starts of by declaring the appropriate variables required for the user to create an event for the application - This is done by using the ‘document.querySelector()’ method which will take the data that the user inputted. Then the program retrieves the data for further use in the script. The getDatabase() function is called from the imports and involves using the Firebase Realtime Database. It initialises a reference to the database and uses the ‘push()’ function to create an entry which will be stored under the ‘/users’ node which will have its own unique identifier (‘userId’). The entry data is declared inside of the push() function containing username, email, location, sport and players number. With this function, a new child node will be initialised every time a new event is created. At the end of the function, once the ‘push()’ has been initialised, the program will automatically refresh using the pre built-in function ‘reload()’ and this will show all the different events on the sidebar, as shown in Figure 16.

Figure 16. User Interface of the Connect page  
A screenshot of a computer

Description automatically generated with medium confidence

Figure 16 is showing the connect page, where users can create an event and see the events they have created within the sidebar. This will strictly show what the user has created but everyone’s events will be shown in the maps page. This is so that the users can be notified and see if there are any updates on whether they have enough players for the event to proceed. Alongside this, you can see a basic format to the event form, which is centred in the middle of the screen, just so that this is user friendly and easy to manage by the users.

Figure 17. Firebase database with the created events  
A screenshot of a computer

Description automatically generated with medium confidence

Figure 17 visually represent the structure of the database and displayed the stored information which was inputted by the user. The node named ‘users’ has got its own child nodes which is the ‘event id’ and inside of that node, will contain the different data that has been stored, as shown in the Figure. These steps will be followed continuously and will always fresh once the user makes another input, or even if a new user has been created.

## Evaluation

All in all, this project was able to follow the steps that was considered when the planning process was taking place. Although this was only a prototype of the anticipated result, the web application itself managed to identify many useful and real-life issues that can be used to help the main issue in which is helping people participate in a variety of sports. Alongside this, the project also managed to apply many different tools such as being able to send data to a database and to also be able to read it, whereby exploring JavaScript and working with Firebase. Firebase is something that hasn’t been around for that long, as the likes of other database management tools have been. Therefore, being able to apply the code and database together and to demonstrate this from the plan, suggests that this was a success to some extent.

On the other hand, some aims didn’t live up to expectation, especially with the interactive maps. This project was planned to be heavily reliant on the interactive map but did not capture everything that was meant to be. One thing that wasn’t implemented that was suggested in the plan was the possibility of having everything clickable on the maps, rather than using an interactive sidebar. In addition to this, instead of having the events on the sidebar, it could’ve just been created into a clickable box with pointers for the users to detect the events on the maps – meaning that the sidebar could’ve been discarded completely. Despite being able to create a routing for two different locations, more could’ve been implemented from the maps side of things.

To conclude, there are some positive things to take from this project but also certain aspects of things that could be done differently if this was meant to be implemented again. Being able to demonstrate an interaction from back-end and front-end was a strength that this project was able to show; it managed to complete a lot of objects that were first initiated in the plan. But the main downside to this was the fact that a completely developed and interactive map wasn’t demonstrated within this project.

## Related Work

Sports player finding applications have gained significant popularity in recent years, offering a convenient way for individuals to connect with like-minded players and participate in their favourite sports. While several existing applications, such as Goals Soccer Centre and other familiar apps, have successfully addressed the player-finding challenge, there is still ample opportunity to explore new avenues and tackle unaddressed issues in this domain. This technology review aims to examine the current landscape of player finding applications, identify potential areas for improvement, and propose innovative approaches to enhance the user experience.

Existing Player Finding Applications:

Goals Soccer Centre and similar apps have demonstrated the value of providing a platform where individuals can search for players, join teams, and book facilities for organized games. These applications often include features like player profiles, skill level filtering, and scheduling tools. They have proven effective in connecting players and promoting community engagement in sports.

Identifying Unaddressed Issues:

While existing player finding applications have made significant strides, there are several unaddressed issues that can be explored to enhance the user experience and provide innovative solutions. Some potential areas to consider include:

1. Improved Matchmaking Algorithms: Current applications rely on basic filtering mechanisms, such as skill level and location, to match players. However, more sophisticated matchmaking algorithms that consider additional factors like playing style, preferred position, and availability could lead to more accurate and compatible player pairings.
2. Real-Time Location Tracking: Integrating real-time GPS tracking capabilities within player finding applications could enable users to locate nearby players in real-time, facilitating spontaneous matches and reducing the time spent on coordination.
3. Social Networking Integration: Incorporating social networking features, such as the ability to follow and connect with favourite players, create sports-related events, and share achievements, can enhance the sense of community and foster ongoing engagement among users.
4. Skill Verification and Endorsements: Implementing a skill verification system, where players can showcase their achievements and receive endorsements from teammates or opponents, can provide a more reliable assessment of their abilities. This would enable users to make informed decisions when selecting players for specific games or teams.
5. Integrated Communication Tools: Including built-in communication tools, such as in-app messaging or voice chat, would streamline the coordination process, allowing players to communicate directly within the application without relying on external messaging platforms.
6. Integration of Augmented Reality (AR): Exploring the integration of AR technology could revolutionize the player finding experience. For example, using AR markers or wearable devices, users could locate and interact with virtual players in real-world environments, enabling immersive and engaging gameplay.

While existing player finding applications like Goals Soccer Centre have paved the way for connecting sports enthusiasts, there are still ample opportunities to explore innovative approaches and address unaddressed issues in this domain. By focusing on areas such as improved matchmaking algorithms, real-time location tracking, social networking integration, skill verification, integrated communication tools, and AR integration, developers can enhance the user experience and revolutionize the way players connect and participate in sports. The proposed enhancements not only seek to improve the functionality and usability of player finding applications but also aim to foster a sense of community, increase user engagement, and provide a platform that caters to the evolving needs of sports enthusiasts. By constantly innovating and pushing the boundaries, these applications can play a crucial role in promoting sports participation, facilitating new connections, and enhancing the overall sports experience for individuals worldwide.

# **Conclusion**

## Overall, my computer science project w… . Despite the initial struggles and personal issues, I faced along the way, I was able to navigate through the challenges with determination and perseverance. Through careful organization and planning, I executed the project effectively, achieving the desired outcomes and exceeding my own expectations.

## From the outset, I recognised the importance of meticulous planning, setting clear goals, and establishing a solid framework for my project.

## This organisational approach proved invaluable as it provided a roadmap to guide me throughout the entire process. By breaking down the project into manageable tasks and setting realistic deadlines, I ensured a structured and efficient workflow. This allowed me to stay focused and make steady progress, even when faced with unexpected obstacles. Furthermore, effective time management played a critical role in the success of my project. I recognised the need to balance my project work with other personal commitments and challenges that arose during the course. Despite these external factors, I remained committed and disciplined, allocating dedicated time slots for research, coding, testing, and documentation. This disciplined approach enabled me to make steady progress and meet important milestones, ensuring that I stayed on track throughout the project's duration. In terms of project execution, I paid meticulous attention to detail, diligently testing and debugging my code to ensure its accuracy and reliability. I sought feedback from my peers and professors, incorporating their suggestions and constructive criticism to refine and improve my work. This collaborative approach not only enhanced the quality of my project but also fostered a supportive and engaging environment that contributed to its success.

## Moreover, I leveraged my personal growth and adaptability to overcome the challenges that arose during the project. Whether it was grappling with complex algorithms, limited resources, or my initial lack of familiarity with JavaScript, I demonstrated resilience and a willingness to learn. Each obstacle became an opportunity for growth, and through persistence and resourcefulness, I was able to overcome these hurdles and produce a robust and functional final product.

## In conclusion, my computer science project stands as a testament to my abilities as a diligent and adaptable student. Through careful planning, effective organization, and a determined mindset, I successfully navigated through the challenges and accomplished the objectives I set out to achieve. This project not only showcased my technical skills but also highlighted my growth as an individual, emphasizing my resilience, problem-solving abilities, and commitment to excellence. I am proud of what I have accomplished and look forward to applying the valuable lessons learned from this project to future endeavours in the field of computer science.

## Reflection

Throughout the duration of my computer science project, I encountered a myriad of challenges that tested both my technical skills and my perseverance. One hurdle I faced was my limited knowledge of JavaScript, which posed a significant obstacle given its crucial role in the project.

As I embarked on the project, I quickly realized that my familiarity with JavaScript was far from adequate. The language presented a steep learning curve, and I often found myself grappling with its syntax, concepts, and best practices. The struggle to comprehend JavaScript added an extra layer of complexity to my project, requiring me to invest significant time and effort into self-study and seeking guidance from knowledgeable resources.

The process of learning JavaScript was not without its difficulties. There were moments when frustration set in as I encountered errors and struggled to grasp certain programming concepts. However, I remained determined and persevered through these challenges. I sought out online tutorials, documentation, and coding forums that provided valuable insights and practical examples. Additionally, I reached out to my professors and classmates for assistance, leveraging their expertise to deepen my understanding of the language.

Despite the initial struggles, I gradually began to gain proficiency in JavaScript. The more I immersed myself in coding and hands-on practice, the more comfortable I became with the language. Each line of code written, and each bug resolved served as a stepping stone toward a greater understanding. This newfound knowledge in JavaScript not only propelled the progress of my project but also instilled a sense of confidence in my ability to learn and adapt to new technologies.

Moreover, the challenges presented by my limited knowledge of JavaScript provided unexpected positive outcomes. They forced me to adopt a growth mindset and fostered a deep sense of curiosity and determination. Rather than shying away from the difficulties, I embraced them as opportunities for personal and intellectual growth. As a result, I not only expanded my expertise in JavaScript but also developed problem-solving skills, critical thinking abilities, and a heightened resilience in the face of adversity.

In retrospect, the struggles I encountered due to my lack of JavaScript knowledge have become an integral part of my project's narrative. They served as a reminder of the importance of continuous learning and adaptation in the ever-evolving field of computer science. The initial difficulties paved the way for personal growth and a deeper appreciation for the intricacies of programming languages.

In conclusion, my computer science project was a journey marked by numerous challenges, among them my limited knowledge of JavaScript. Overcoming these struggles required dedication, perseverance, and a commitment to continuous learning. Through hours of self-study, seeking assistance from knowledgeable resources, and hands-on practice, I was able to navigate the complexities of JavaScript and successfully implement it in my project. The experience not only deepened my technical skills but also fostered personal growth, resilience, and an unwavering determination to conquer future obstacles in my computer science journey.

## Future Work

The aim of this computer science project was to develop a prototype that incorporates inclusive design principles, focusing on providing features for different demographics, such as an older audience and individuals that may have a disability. Since the current state of the implementation is shown to be a prototype, there is a need for future work to expand and refine the project to make it more accessible and inclusive for the expanded demographic. For example, in order to manage this project and allow it to be accessible for the older and disabled individual, some future work can involve conducting more user research and integrating the individuals feedback. The research can consist of the challenges they may face within the sporting community, or even a case of where it may be a bit difficult for them to use computer systems for example. Gathering this feedback will allow me to gain a great insight from the focused demographic and allow me to expand more on the different tools I may need to use in order to give the target users, a comfortable and easy experience when using the project – an example would be is that for the users that may have a disability, they can have an assistant option which can help them access different features. These are tools that are already implemented with other different applications whether it’s on a mobile phone or a desktop, so this would be something I would look closely towards in future works. These technologies should be explored and integrated into the project to empower users with diverse abilities.

Alongside the necessary tools to help different demographics, there could be other things I could’ve chosen to do different alongside this. From a visual aspect side of things, features such as font size or the colour of different things can be a good way to make the project more user friendly. Therefore, allowing the users to be able to be freer with the application or from a visual side of thing, they could have disabilities too such as colour blindness so this could tackle their individual challenges. Simple adaptations like this could make this experience better even for just the one user.

Another aspect I would’ve liked to have changed is the approach on how I layered each page. I did have fun trying to explore different techniques and layouts with this web application, but I feel as if the buttons for example, could have a little more of a lift and I feel as if I could’ve enhanced the animations instead of just using the hovering effect. This would have made the project more professional and maybe a little more eye catching for the users. However, since this is a prototype, this is something, I could also think about for future works.

One of the key things I would like to take more into consideration would be from the ethical side of things and maybe take into consideration protecting users’ data and privacy. Handling a user’s information securely is crucial and definitely something I must take into action for future works, and follow the guidelines of keeping everything lawful, i.e. not handing out key information of the users. Instead of asking for the user’s postal code, maybe just ask for a region of where they would like to play in order to tackle these challenges.

To conclude, there are a lot of different considerations to take into account for future works. As said before, since this is a prototype, a lot of work can be done to enhance the inclusiveness and accessibility. Mainly trying to focus on the disabled and older individual as I feel like there’s a lot of room for these two demographics and I feel as if it can be done successfully with the help of data research, designing and even getting feedback from the sports enthusiasts from these two individual demographics. This would help with allowing a wider range of audience to be using this application.

# **References**

In this section, you **must** reference any sources used in your work. Typically, these sources will have come up during the investigation and related work sections. Your referencing must use the IEEE referencing style [IEEE Citation Guidelines2.doc (ieee-dataport.org)](https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf) .

It is **highly** recommended that you use reference management software such as Mendeley or Zotero.

<https://randomnerdtutorials.com/esp32-firebase-web-app/#install-software>

<https://webdeasy.de/en/top-css-buttons-en/>

<https://stackoverflow.com/questions/66879378/typeerror-undefined-is-not-an-object-evaluating-app-firestore>

<https://stackoverflow.com/questions/69230383/failed-to-resolve-module-specifier-firebase-app>

[1] https://firebase.google.com

[2] <https://successive.tech/blog/best-database-for-web-applications/>

[3] https://www.youtube.com/watch?v=rQvOAnNvcNQ

[4] github wireframe maps link

[5] github wireframe log in page link

[6] https://www.goalsfootball.co.uk/

# **Appendices**

Appendices appear after references. Your appendices depend on the nature of your project. **Do not assume people will read your appendices.** Even if you direct them to do so in your main text, appendices are considered additional information and should not be relied upon to understand your main body of work. Refer readers to an appendix using a phrase such as *see Appendix A for further details*.

The following documents **must** be included as references:

* Your Project Proposal.
* Your Progress Review Form.
* Your original plan and revised plans as your project evolved.
* A description of how to access any technical output. **It is strongly recommended you use GitHub or something similar to do this.**

Any important communications between you and external stakeholders -- **please ensure private data is removed and communications anonymised.**