Description of a Problem

I intend to develop a football draft website in which the user can create and view a team from a selection of the best players of all time, create a club with a name of their choice and compare their team with other users. I intend to include the following features: a login/register system, a help screen, as well as the ability to view other user’s teams in a HTML table, and their own team on a pitch image (with images of selected players) after completion. The user will select a player for each position, with the players displayed as a list to select from in a HTML table populated by a database. The end-users of the game will be avid football fans who want to compare their teams with friends and others to discuss their teams’ differences. My project meets the advanced higher computing requirements because it will have a user-friendly user interface, with validated inputs. My project will also include a website integrated with a database storing the footballer details, user team details as well as the user details, with SQL being used to update and alter these tables when required for the game to function. Media queries will also be utilised in order to change the appearance of pages to suit different sizes of screen displays and to further meet the requirements of the AH Computing Science course as PHP is used throughout in order to create session variables and create the link to and from the website and database.

|  |  |
| --- | --- |
| Advanced Higher Techniques being met |  |
| Use of media queries to suit screen size |  |
| Website integrated with external database, with validated inputs |  |

Feasibility Study

Economic – The cost will be nothing to make this project as all software’s and resources needed are provided by the school, making the project cost effective for myself as it allows me to show and develop the skills I have learnt over the AH course.

Time – I will be given until the end of March to complete my project so I need to ensure to plan my time appropriately and to include all the features that are in my analysis and design.

Legal – I will have to make sure to co-operate with legal laws such as GDPR, but one concern may be the use of footballer images in my database, breaching copyright laws. As this is a school project and after consulting with my teacher, I will be using these images without seeking permission from the owners of the images. If I were to use these images for commercial purposes I would need to seek permission with regards to copyright protection and image rights with regards to the players. I also need to ensure I store the user’s personal data such as email, age and phone number securely and only use it lawfully, when they use it to register on the registration form or else I could risk breaking GDPR.

Technical – Like the economic aspect, everything needed to carry out my project is supplied by the school as well as I myself owning a desktop PC at home with the necessary softwares needed installed.

**Scope**

The scope of my project will include:

1. A completed design with a data dictionary, use case diagram, entity-relationship diagram as well as wireframes for each screen interface.

2. A functional football draft website, with 3 integrated databases, with necessary validation and existing data.

3. A completed comprehensive test plan including a test table with expected outputs and results

4. Log of ongoing testing

5. An evaluation of the fitness for purpose as well as a summary of new skills gained

**Constraints**

There is a number of economic, technical and time constraints that will occur during this development –

1. I must ensure I complete my project by the deadline of the 23rd of March, so it is complete in time for SQA marking.

2. I will use software made available within in school, this being Notepad++ and a XAMPP web server to store my database. This means there is no cost.

3. Other users will be able to access my school computer so I need to maintain the safety of my projects files and regularly back them up to another accessible location.

4. I will be learning PHP during the AH course, meaning that it will be a new language to me.

5. I must make sure to comply with the Copyright, Design and Patents Act and ensure that the GDPR Regulations are met.

**Boundaries**

1. Registered accounts can not have the username of an already existing account

o Users logging in must do so with matching credentials of an existing account

2. User can’t create more than one team at once

3. the team is only displayed in a 4-4-2 formation

4. Each position can only be selected from an existing list of players

o the user can’t add substitutes

o the user can’t add new players that aren’t already in the footballer database

5. The eleven players selected by the user are stored in a ‘userteam’ table with the user’s username as a field, so 11 players can be associated with the username.

6. Usernames will be displayed in ascending order when choosing one to compare.

**Requirements Specification**

The following requirements are required in order to carry out the purpose of the website; a football draft website accessible after the creation of and logging in of an account in which the user can create their own team and view teams created by other users -

**Functional Requirements**

· The website should be able to access and store details of users and footballers as well as their team in an external database.

· Easy to navigate and follow for the ease of the user, with instructions clear throughout

· Ensure only logged in users can access the full website

· Search for and view other user teams by connecting to the external database and conducting SQL queries

* Pages of website formatted with external stylesheet
* User should be able to compare their team to another user’s team

· All user inputs should be validated

· Responsive screen layout, using media queries. Apart from the home page, a navigation bar should be present on all pages which then turns into a collapsible side bar with a menu button when screen size reaches 800px.

* Session variables must be used to store the user’s login credentials on login throughout the website.
* View existing team on a pitch image with player names and images in a 4-4-2 formation
* User creates their team from dropdown lists in a html table

**End-user requirements**

Through conducting user surveys with 5 people most likely to use my website I found that users of the football draft website should be able to:

* Navigate the website easily, thanks to meaningful instructions, error messages and layout of pages
* Create an account and log in
* Find the appearance of website simple and easy to follow, whilst being visually pleasing.
* Be able to create own team
* Be able to view their team when completed
* Be able to compare their team with another user of their choice

**UML Use Case Diagram**

**Diagram

Description automatically generated**

Compare team to other users

**User Inputs**

1. Register details (username, password, clubName, favteam, age, email, phonenumber)
2. Userteam (selected from dropdown list of footballers)

**Website Inputs, Processes and Outputs**

**Inputs**

* Size of browser window
* footballer , userteam, userdetails tables
* Login query results
* Registration query results

**Outputs**

* Error messages
* Login messages, and login status using session variables
* Registration messages – for successful/unsuccessful registration
* Search results for other user teams
* Finished user team with images and name on pitch
* How it works page

**Processes**

* Opening and closing connection to database
* Validation of user inputs
* Store user team and user details in database
* Populate html tables/drop down lists with data from database
* Utilise session variables to use personal user info across web pages
* Execution of SQL queries
* Place user team on pitch using multi dimensional array of positions

**Project Plan**

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Est total Duration (hours)** | **Target Date** |
| Analysis   * Come up with initial project idea that meets AH requirements * Write outline of project * Constraints boundaries and scope documented * Functional requirements * End-user requirements * Inputs, Processes and Outputs described * Creation of Project Plan   Total days: 21 | **6** | 24th October |
| Design   * Wireframes for each screen * Pseudocode for server-side tasks (integration with database) * Data Dictionaries for database   Total days: 21 | 6 | 14th November |
| Implementation   * Formation of databases * HTML pages created * CSS styling to pages * PHP (registration) * PHP (login) * PHP (session variables) * PHP (creation of user team and process of populating html tables with database data) * Media queries * Validation (registration and login process) * Validation (only 11 footballers can be on user team) * Ongoing testing * Description of research and new skills learned outside of AH course.   Total days: 107 | 35 | 1st March 2023 |
| Testing   * Comprehensive test plan * Requirements testing * Test Cases   Total days: 21 | 10 | 22nd March 2023 |
| Evaluation   * Fitness for purpose evaluation * Future maintainability * Robustness   Total days: 7 | 6 | 29th March 2023 |

**Resources Required**

The following resources are needed for each stage of the development:

|  |  |
| --- | --- |
| Analysis | * Microsoft Office Word 2016 * Google Chrome |
| Design | * Microsoft Office Word 2016 * Google Chrome |
| Implementation | * Notepad++ * XAMPP server * Microsoft Office Word 2016 * Google Chrome |
| Testing | * Microsoft Office Word 2016 * Google Chrome * XAMPP server |
| Evaluation | * Microsoft Office Word 2016 |

**Project Design**

**Pseudocode for server-side processes**

**Registration Process**

1. Start session
2. Assign connection details to PHP variables
3. Connect to database
4. If there is an error with connection then
5. Display error message
6. End if
7. Assign values submitted by HTML registration form to PHP variables using $\_POST
8. Create and execute SQL query to search existing userdetails table for any already existing accounts with a username the one stored in the PHP username variable.
9. If query returns more than 0 rows Then
10. Display error message stating that there is an account already with the chosen username, with link to register/login page.
11. Close connection
12. Else
13. Create and execute query to add user details to user details table
14. End if
15. Close connection

**Login Process**

1. Start session
2. Assign values submitted by HTML registration form to PHP variables using $\_POST
3. Assign connection details to PHP variables
4. Connect to database
5. If there is an error with connection then
6. Display error message
7. End if
8. Create SQL query to search for entered username in database
9. Execute SQL query
10. If a row is returned and password stored in userdetails table = password entered Then
11. session variables are assigned
12. Display personalised welcome message
13. Redirect to home page
14. End connection
15. Else
16. End connection
17. Display message saying unknown username/password
18. Redirect to login page
19. End if

**Display of footballers in dropdown list in HTML table**

1. Start session
2. Assign connection details to PHP variables
3. Connect to database
4. If there is an error with connection then
5. Display error message
6. End if
7. Create SQL query to only select the all footballers from footballer table
8. Execute SQL query
9. Set query results to an array called $footballer holding playerName and position
10. Loop for each row in HTML table
11. Start switch statement
12. Case 1, gk, case 2,3,4 defenders, case 6,7,8,9 midfielders, case 10 and 11 attackers.
13. Insert footballer into appropriate rows for drop down list based on their position in $footballer array
14. End switch statement
15. Close connection

**Logout process ( logout.php)**

1. Start session
2. Unset all session variables
3. Destroy the session
4. Redirect to welcome page

**Duplicate check for submission of user team, and insertion of userteam**

1. Start session
2. Set database connection values to PHP variables
3. Connect to database
4. If there is an error with connection then
5. Display error message
6. End if
7. Get size of $\_POST array (should be 33 for 11 players with 3 columns)
8. Divide size by 3 to get the number of players
9. Loop for each player
10. Retrieve information for each player from $\_POST array
11. Check for duplicates entered by the user by running a SELECT query for previous player names in array
12. If there are duplicate entries, display an error message and link
13. Delete the duplicate entries
14. Exit the script
15. End if
16. End loop
17. If there are no duplicate entries, insert the HTML table row into the userteam table by running an INSERT query
18. If $i is equal to the number of players (all players inserted) use JavaScript to show a success message and hide the contents of the page
19. End if
20. End if

**Delete team if user tries to make a new one when user has already**

1. Start session
2. Set database connection values to PHP variables
3. Connect to database
4. If there is an error with connection then
5. Display error message
6. End if
7. If user clicks begin button
8. Create and execute query to delete any players belonging to current user in userteam table
9. Redirect user to build team page
10. End if
11. Exit script

**Display user team on pitch image**

1. Start session
2. Set database connection values to PHP variables
3. Connect to database
4. If there is an error with connection then
5. Display error message
6. End if
7. Create and execute query to select user’s 11 players from userteam
8. Add players to an array called $footballers array
9. Close connection
10. If $footballers does not have 11 footballers then
11. Display message saying you have not made a team yet
12. Redirect to homepage
13. End if
14. Create a multi-dimensional array called $positions holding an x and y value
15. Loop through each footballer in array and allocate positions with corresponding loop values and top: (y value)px; and left: (x value)px;
16. Add image of footballer using the image field stored in userteam table and image stored in htdocs folder
17. Add corresponding player name underneath

**Populate dropdown list for footballers on build team page**

1. Start session
2. Set database connection values to PHP variables
3. Connect to database
4. If there is an error with connection then
5. Display error message
6. End if
7. Create and execute query to select user’s 11 players from userteam
8. If query does not return 11 results
9. Display message saying you have not made a team yet
10. Redirect to homepage
11. End if
12. Create and execute query to find all usernames in userteam table that is not the current user and display each one as a selectable drop down list
13. When user selects their username, POST the username to viewother.php

**Display userteam alongside chosen username’s team**

1. Start session
2. Set database connection values to PHP variables
3. Connect to database
4. If there is an error with connection then
5. Display error message
6. End if
7. Set current user’s username to php variable
8. Create and execute query to retrieve receive current user’s userteam
9. Populate html table with current user’s team
10. Set the username of the user being compared to php variable using POST from when user selected username from dropdown list
11. Create and execute query to retrieve userteam of the chosen user
12. Populate html table with userteam of chosen user

* **Adv higher skills in analysis**
* **Pseudocode**
* **Erd diagram**
* **Checklist for requirements for fitness for purpose**

**Data Dictionaries**

Userdetails table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Field Length/Type | Required | Key |
| username | VARCHAR(255) | Yes | Primary |
| Password | VARCHAR(255) | Yes |  |
| clubName | VARCHAR(255) | Yes |  |
| favteam | VARCHAR(255) | Yes |  |
| age | INT(3) | Yes |  |
| email | VARCHAR(255) | Yes |  |
| phonenumber | VARCHAR(11) | Yes |  |

Footballer table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Field Length/Type | Required | Key |
| PlayerCode | INT auto increment | Yes | Primary key (auto increment) |
| playerName | VARCHAR(255) | Yes |  |
| position | VARCHAR(255) Restricted to: (`GK`,`Defender`,`Midfielder`,`Attacker`) | Yes |  |

Userteam table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Field Length/Type | Required | Key |
| playerName | VARCHAR(255) | Yes |  |
| position | VARCHAR(255) | Yes | Restricted choice |
| number | VARCHAR(255) | yes | PK (compound key) |
| username | VARCHAR(255) | yes | PK (compound key),FK |
| image | VARCHAR(255) | yes |  |

**Hierarchical Design of Website Structure**

Diagram

Description automatically generated

**Query Design**

Throughout my website I plan to utilise numerous queries, the query designs are shown below with a brief description of what they are to be used for.

**Registration Duplicate check Query**

|  |  |
| --- | --- |
| **SELECT** | All fields (\*) |
| **FROM** | userDetails |
| **WHERE** | Username = username entered |

**Registration Query**

|  |  |
| --- | --- |
| **INSERT** | Username, password, clubName |
| **TABLE** | userDetails |
| **VALUES** | Username, password, clubName |

**Login Query**

|  |  |
| --- | --- |
| **SELECT** | All fields from userdetails (\*) |
| **FROM** | userDetails |
| **WHERE** | Username = username entered |

**Delete current user’s old team if they had previously created one**

|  |  |
| --- | --- |
| **DELETE FROM** | userteam |
| **WHERE** | Username = username of current user |

**Collection of footballers from database to be stored in an array, to be used to populate dropdown list of players (in html table)**

|  |  |
| --- | --- |
| **SELECT** | All fields (\*) |
| **FROM** | footballer |
| **ORDER BY** | PlayerName |

**Check for duplicate entries from user for user team (when user submits 11 players from their team)**

|  |  |
| --- | --- |
| **SELECT** | All fields (\*) |
| **FROM** | userteam |
| **WHERE** | Username = current user’s username AND playerName = any playername in current user’s userteam |

**The following query will be used in two instances to:**

* **Delete current user’s old team if they had previously created one and are creating a new one**
* **Delete current user’s team if duplicate entries are submitted when creating team on**

|  |  |
| --- | --- |
| **DELETE FROM** | userteam |
| **WHERE** | Username = username of current user |

**Inserting new team into userteam table**

|  |  |
| --- | --- |
| **INSERT** | PlayerName, position, number, username, image |
| **TABLE** | userdetails |
| **VALUES** | playerName in current loop of html table, position in current loop of html table, username in current loop of html table, playername in current loop of html table concatenated with “.jpg” |

**The following query will be used in two instances to:**

* **Find the current user’s team to be displayed with images on pitch in formation**
* **Find current user’s team to be displayed to screen in html table**

|  |  |
| --- | --- |
| **SELECT** | All fields (\*) |
| **FROM** | userteam |
| **WHERE** | Username = username of current user |

**Find the clubName of the username that the current user selected, to display to screen (viewother.php)**

|  |  |
| --- | --- |
| **SELECT** | All fields (\*) |
| **FROM** | userdetails |
| **WHERE** | Username = the username selected by the current user (on selectother.php) |

**Find the team the current user wanted to view (viewother.php)**

|  |  |
| --- | --- |
| **SELECT** | All fields (\*) |
| **FROM** | userteam |
| **WHERE** | Username = the username selected by the current user (on selectother.php) |

**Find users with a team created, not including the current user – this is to populate the dropdown list on selectother.php**

|  |  |
| --- | --- |
| **SELECT** | username |
| **FROM** | userteam |
| **WHERE** | Username NOT the username of the current user |
| **GROUP BY** | username |

**Diagram

Description automatically generatedUser interface wireframes**

**A picture containing diagram

Description automatically generated**

**Diagram

Description automatically generatedDiagram

Description automatically generated**

**A picture containing diagram

Description automatically generatedDiagram

Description automatically generated with medium confidenceDiagram

Description automatically generated**

On login, session variables are set for the user to be used throughout the website

**Text

Description automatically generated**

**Graphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceMedia query for full website pages apart from home page**

**Table

Description automatically generatedDiagram

Description automatically generated**

**Diagram

Description automatically generatedDiagram

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

Graphical user interface

Description automatically generated

When username is selected and compare is clicked, the username chosen is stored using POST, to be used to conduct the query on the following page to display their team.

Table

Description automatically generated

**Media query design**

**The design for the media query shown previously in the user interface screenshots from horizontal navbar to collapsible sidebar with a menu button is as follows –**

**Larger than 800px display :**

|  |  |
| --- | --- |
| **.menu** | **Display:none** |
| **.navbar** | **Width: 100% - and all styles applied** |
| **.menu icon** | **Display:none** |

**Larger than 800px display :**

|  |  |
| --- | --- |
| **.menu** | **Display:block – width: 25%, height 240px – and all styles applied** |
| **.navbar** | **Display:none** |
| **.menu icon** | **Display: block** |

**This media query allows all pages to be accessible from a wide variety of screen sizes. As it has been used across multiple pages I will use an external stylesheet to store the CSS for this media query. I will also two other external stylesheets to change the colour and font of the webpages as well as to enable the use of icons – I got this style sheet from w3schools. I also plan to use media queries to reduce fonts and other elements in the page to suit the screen size.**

**https://www.w3schools.com/icons/tryit.asp?filename=tryicons\_fa-sign-out**

**Diagram

Description automatically generatedGraphical user interface

Description automatically generated with low confidenceFirstPage.html - Implementation**

Graphical user interface

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated

**RegisterPage.html/register.php**

Graphical user interface

Description automatically generatedScreenshot of userdetails table after registration of exampleaccount

**Text

Description automatically generatedGraphical user interface, text

Description automatically generated**

**Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated**

**LoginPage.html/login.php**

**Graphical user interface, text

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, website

Description automatically generatedGraphical user interface, text, application, website

Description automatically generated**

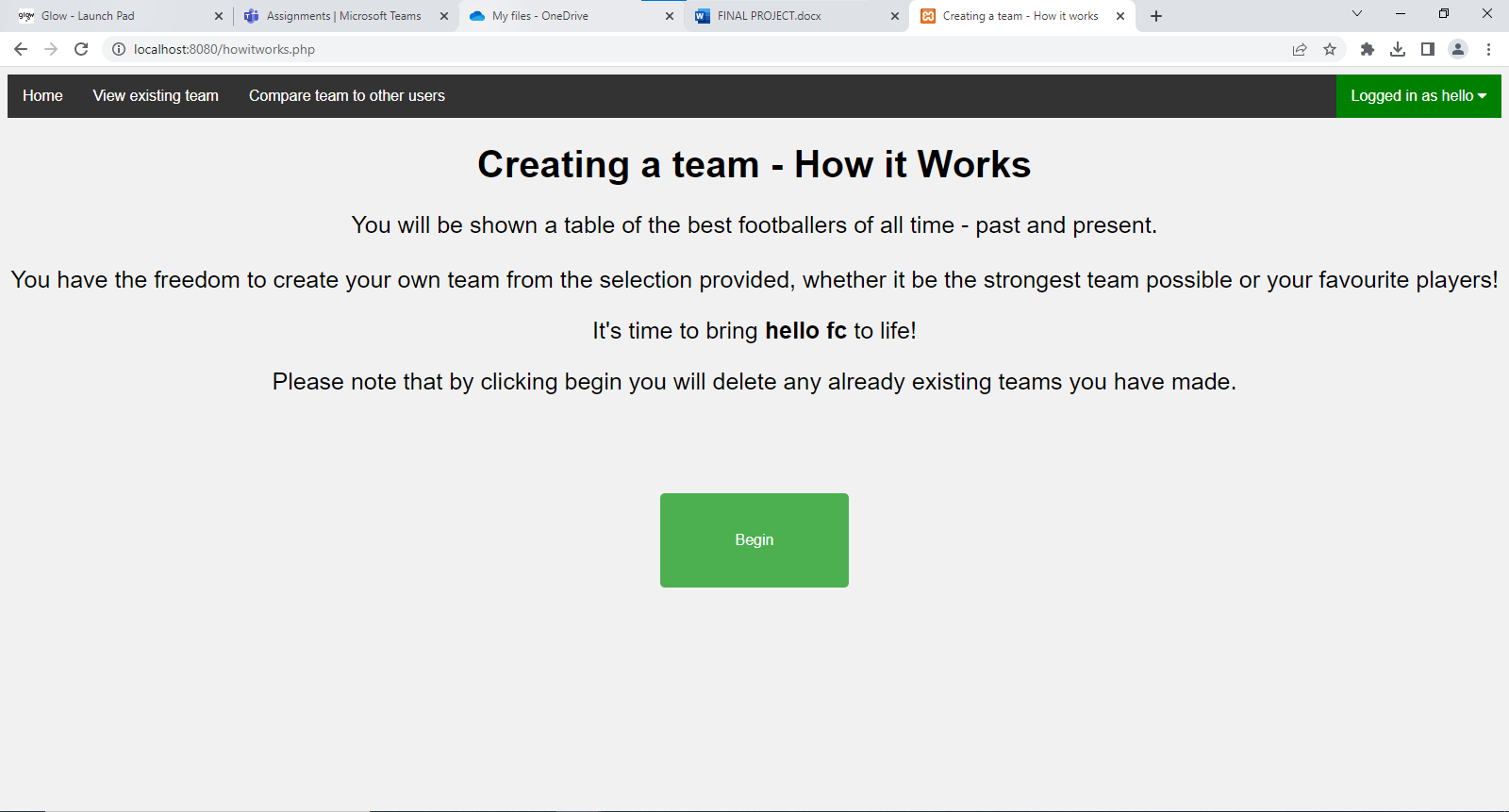
**HomePage.php**

**Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

**Howitworks.php**

****

**buildteam.php**

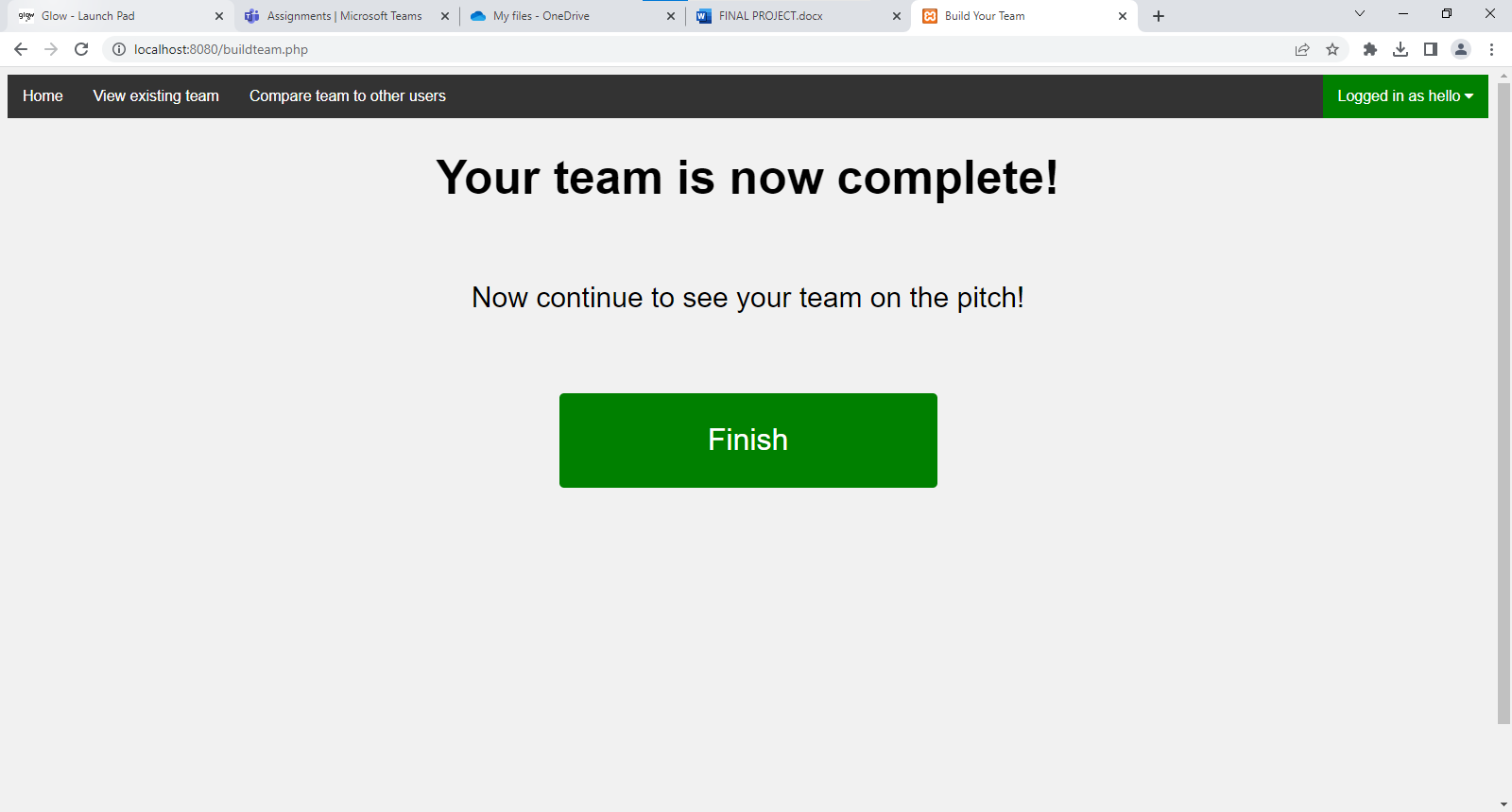
**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated**

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

**finishedteam.php**

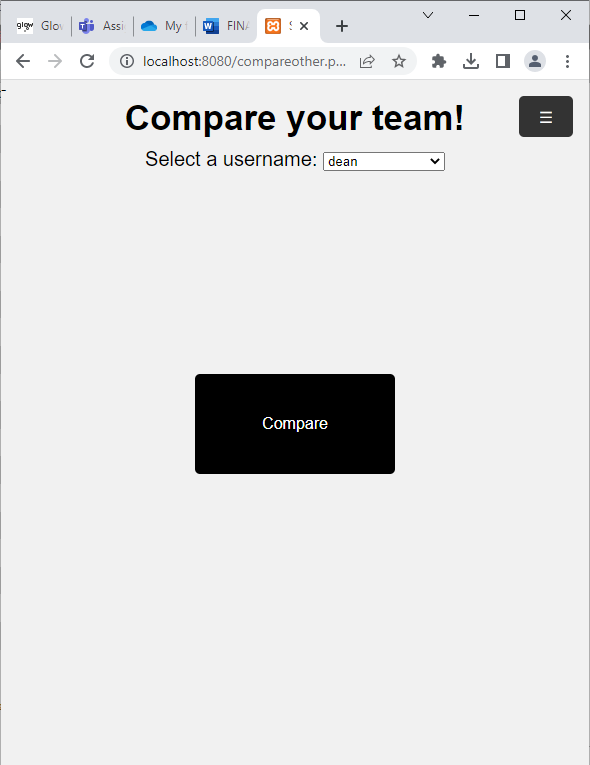
**Graphical user interface, application

Description automatically generated**

**A picture containing map

Description automatically generatedA screenshot of a game

Description automatically generated with medium confidence**

**Graphical user interface, application

Description automatically generated**

**Compareother.php**

****

**viewother.php**

**A screenshot of a computer

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Log of ongoing testing**

|  |  |
| --- | --- |
| **Problem** | **Solution** |
| **On submission of registration, phone number always being entered as** 2147483647 | **Change phone number data type from INT(11) to VARCHAR(11)** |
| **Player photos not showing** | **Ensure all photos are stored as jpg, and haven’t been automatically converted to another format** |
| **2 Error messages being shown when only one needs to be shown for duplicate entry of username** | **Use ini\_set('display\_errors', 0); to hide the database error and instead only the custom error message is displayed.** |
| **Media query not showing menu icon, only checkbox** | **Missing closed bracket in stylesheet for nav bar** |
| **Register success message not showing** | **Connection was being closed at the end of if statement for duplicate username when it wasn’t needed.** |
| **CSS styles not updating for media query** | **Ensure that the element being styled is referred to in the same way – e.g. in the main CSS, #example h1 must be then followed by #example h1 in the media query rather than just ‘h1’.** |
| **When submitting the user’s team either 2 buttons would have to be used ( submit and then finish to go to next page) or the validation would be skipped by using action = finishedteam.php** | **Use javascript to hide all the initial elements on the buildteam.php page to give the illusion that a new page has been loaded which is actually just a success message with a button to view their new team. This eliminates confusion for the user as they now only have to click one button at a time instead of 2 on their screen at once.** |
| **Logout button wasn’t ending session, and in attempts to implement end session, the session would be ended when the page loaded** | **Make a new php page called logout that the user is redirected to which ends session, unsets all session variables and redirects user to the welcome page.** |

I also ran in to other minor errors such as missing syntax but decided that these errors were not worth documenting.

**New skills researched and developed**

1. **Storing and Processing Image Details**

When users view their team I wanted a picture of the players to appear. After visiting a website that said the simplest way to do this was to save the image path I figured I could store the address of the image in the image field in the user team table, and make the image field the playername concatenated with .jpg. As long as the photo names were saved in my htdocs file with the correct name it worked. I also had to ensure that the images were all 16:9 aspect ratio so they would appear properly within the size of the player element. If I were to develop the project further I would have to think about copyright laws and if I would have to gain permission to use these footballer images for use.

1. **PHP switch statement**

Initially I tried to populate the html tables with if statements but couldn’t get the correct players read into the drop down list, I did some research to find out what else I could do to differentiate what populated each cell for the positions, after doing some research I found that switch statements would allow me to do this for all columns on buildteam.php. It allowed me to change what players were available in each drop-down list, what position was displayed in column two and the shirt number for each row.

1. Arrays

I had to use arrays in numerous places to be able to loop for each player and able to temporarily store the players throughout the pages. I had to make use of an array to ensure that the correct position of players populated the dropdown list. I thought that using an array would be more efficient as repeating queries and looping more times for footballer details would result in more code making the program less efficient. It also made it easier for me to visualise what was happening in the program.

1. Multidimensional Arrays

After doing some research I learned that if I wanted to store x and y values for the players to be displayed on I would have to use a multidimensional array as I had to store more than one key, this then allowed me to enter values for the top and right px values for the player elements on the pitch.

1. Show/hide password feature

After conducting independent research, I found a way to make a button to show and hide the password in the text box for the registration and login pages. This helps the user ensure they are entering the password they intended to.

**Resources used**

**1)** **https://mysql.tutorials24x7.com/blog/how-to-store-pictures-in-mysql-database**

**2)** [**https://www.w3schools.com/php/php\_switch.asp**](https://www.w3schools.com/php/php_switch.asp)

**3)** [**https://www.w3schools.com/php/php\_arrays.asp**](https://www.w3schools.com/php/php_arrays.asp)

**4)** [**https://www.w3schools.com/php/php\_arrays\_multidimensional.asp**](https://www.w3schools.com/php/php_arrays_multidimensional.asp)

**5) https://www.w3schools.com/howto/howto\_js\_toggle\_password.asp**

**Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case ID** | **Test case objective** | **Test case description** | **Expected result** | **Actual Result** |
| 1 | Check communication from login and registration forms on website to the database server. | Enter all registration details and login details as well as details that should not be accepted. | Appropriate message displayed for the user input (successful login or registration/duplicate username entered/wrong login) indicating that either the user’s details has been added to the database, validated as an existing user or requested to be re-entered. | Operates as expected |
| 2 | Check that the logged in user’s details has been stored in a session variable | Log in with an existing account using username and password. | The account username should be displayed in the navbar on certain pages as “Logged in as :” as well as personalised messages throughout the website on multiple pages for example the “How it Works Page” where the clubName of the user should be shown. | Operates as expected |
| 3 | Check that user’s with no team created can’t view their team or compare it to others | Try to select view existing team and compare team to others on an account that have not yet made a team. | Message displayed saying that you must make a team before viewing or comparing your team. | Operates as expected |
| 4 | Check that when the user clicks “begin” to create their team that any existing team has been deleted. | Click Begin on the How it Works Page. | After the user clicks begin, and then navigates to view/compare team on the navbar/sidebar without creating their team, it should read the error message that the user needs to create a team.  Also, a database error would be displayed if there was already a team when another is submitted as username and number is a compound key, and the number only range from 1 to 11 (as seen on the table in the html table on buildteam.php) this should not happen, so the expected result is to see buildteam.php after clicking the begin button, and then the success message to say the team has been made. | Operates as expected |
| 5 | Check that the when the user’s team is created it is stored in the database | Create a team and then navigate to view finished team or compare team page | The players that were selected by the user are displayed in their correct positions on the pitch on the view team page and table in the compare team page. | Operates as expected |
| 6 | Check that the created team can be viewed by other users | Create a team on one account 1 and then log in to account 2 to create another team and see if you are able to compare account 1 and 2’s teams. | When finishing the creation of account 2’s team they should be able to see account 1 in the drop down list of Select a username, when going to the compare team pages, and then proceed to see their team | Operates as expected |
| 7 | Ensure that the footballers have populated the html table correctly on the build team page | Inspect the table and see if players are in the correct cell location based on their football position. As well as the correct number of footballers are present. | The different unique footballer names can be counted in the dropdown list for the html table and this number should equal the number of rows in footballer database table. The developer of the project will also be able to tell if footballers are in the right position based on general football knowledge. | Operates as expected |
| 8 | Check that information in the registration page is validated by the form. | Attempt to enter phone numbers with letters/longer than 11 numbers and email with incorrect format. | On submission a message is displayed on the form page. | Operates as expected |
| 9 | Ensure that the validation for duplicated player entries for creating a team works | Create and submit a team with the same player selected in multiple positions | Error message is displayed saying there are duplicate entries, and user is redirected back to build team page | Operates as expected |
| 10 | User interface is suitable for desktop and mobile device sized screens | Adjust window size to see if the website is still navigable and easy to use. | Media queries are executed to alter web page interface to match the user’s screen size. | Operates as expected |

|  |  |
| --- | --- |
| Test Case ID | Evidence of result |
| 1 | Text  Description automatically generated |
| 2 | Graphical user interface, text, application, email  Description automatically generated |
| 3 | Graphical user interface, application  Description automatically generated |
| 4 | A screenshot of a computer  Description automatically generatedGraphical user interface, application, Teams  Description automatically generated |
| 5 | **Graphical user interface, application  Description automatically generated** |
| 5 (screenshot 2) | **A screenshot of a computer  Description automatically generated** |
| 6 | Graphical user interface, text  Description automatically generated |
| 6 (screenshot 2) | A screenshot of a computer  Description automatically generated |
| 7 | Graphical user interface, application  Description automatically generated  Snippet from this test – 6 goalkeepers in footballer table  6 goalkeeper names in dropdown list – the same result happened for each defence, midfield and attacking position |
| 7) screenshot 2 | A screenshot of a computer  Description automatically generated  6 goalkeeper names in dropdown list – the same result happened for each defence, midfield and attacking position |
| 8 | Graphical user interface, text, application  Description automatically generatedApplication  Description automatically generated with low confidence  Graphical user interface, application  Description automatically generated |
| 9) screenshot 1 | Graphical user interface, table  Description automatically generated |
| 9) screenshot 2 | Graphical user interface, text, application  Description automatically generated |
| 10) | All media queries function to the ensure content fits screen size well, as seen in screenshots of the website in the implementation. |

**Comments from testing**

Test case 3– On testing I made a tweak to redirect to the homepage rather than the how it works page.

**Requirements testing**

**I asked the same users who did the user survey for the end user requirements if they thought that each requirement had been met when testing the finished website.**

|  |  |
| --- | --- |
| **End user requirements** | **Met** |
| * Navigate the website easily, thanks to meaningful instructions and error messages |  |
| * Create an account and log in |  |
| * Find the appearance of website simple and easy to follow , whilst being visually pleasing. |  |
| * Be able to create a team |  |
| * Be able to view their team when completed |  |
| * Be able to compare their team with another user of their choice |  |

5 out of 5 users agreed that these had been met.

**Functional requirements test – bold if met, with explanation**

· The website should be able to access and store details of users and footballers as well as their team in an external database – **The website uses these 3 tables in an external database called project, and accesses the tables throughout the website through using PHP and SQL queries.**

· Easy to navigate and follow for the ease of the user, with instructions clear throughout – **The website has clear instructions and prompts such as meaningful button text and titles as well as instructions when needed such as “instructions and info” on buildteam.php or the How it Works page, aided by error messages for the user to stay on the right track. The horizontal navigation bar and sidebar media query allows for quick movement between pages in the website too.**

· Ensure only logged in users can access the full website – **Users must undergo the login and registration process before being able to access the homepage and then the rest of the website**

· Search for and view other user teams by connecting to the external database and conducting SQL queries – **When comparing the user’s team they are able to choose another user to compare their team with. This is populated by the userteam table and validated to ensure that only users with 11 players in their team are shown.**

* User should be able to compare their team to another user’s team – **When user selects the username of their choice the other user’s team is displayed on the right side of the screen with the heading of the other user’s club name and a sentence saying it’s this user’s team.**
* Pages of website formatted with external stylesheet – **3 external stylesheets are used throughout**
  + **Mystyles.css - general theme of the website (font-family, background colour)**
  + **Navbar.css - navigation bar and sidebar query**
  + [**https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css**](https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css) **- used for icons for sign out etc, found on w3schools - https://www.w3schools.com/icons/tryit.asp?filename=tryicons\_fa-sign-out**

· All user inputs should be validated – **Registration and login processes both validated by the form (field length and appropriate format) and server side processes (to check if credentials were a match to a record in userdetails or a duplicate username if registering a new account)** .

· Responsive screen layout, using media queries. Apart from the home page, a navigation bar should be present on all pages which then turns into a collapsible side bar with a menu button when screen size reaches 800px. – **This has been implemented exactly as stated in the requirements.**

· Session variables must be used to store the user’s login credentials on login throughout the website. – **session variables have been used to frequently to store the current users username and clubName such as for display personalised messages such as “logged in as” on the navigation bar and Introducing clubName on the finished team page.**

* View existing team on a pitch image with player names and images in a 4-4-2 formation – **The user sees their team in a 4-4-2 formation due to positional values being stored in a multi-dimensional array to hold 2 values. The pitch is also rotated and position of players altered to fit a smaller screen size.**
* User creates their team from dropdown lists in a html table populated by footballers in a footballer database table using SQL queries – **Each dropdown list is populated by players from the appropriate position (GK, Defender, Midfielder, Attacker) to ensure the correct selection of players is available for each row.**

**To summarise :**

|  |  |
| --- | --- |
| **Functional Requirements** | **Met** |
| The website should be able to access and store details of users and footballers as well as their team in an external database. |  |
| Easy to navigate and follow for the ease of the user, with instructions clear throughout |  |
| Ensure only logged in users can access the full website |  |
| Search for and view other user teams by connecting to the external database and conducting SQL queries |  |
| Pages of website formatted with external stylesheet |  |
| User should be able to compare their team to another user’s team |  |
| All user inputs should be validated |  |
| Responsive screen layout, using media queries. Apart from the home page, a navigation bar should be present on all pages which then turns into a collapsible side bar with a menu button when screen size reaches 800px. |  |
| Session variables must be used to store the user’s login credentials on login throughout the website. |  |
| View existing team on a pitch image with player names and images in a 4-4-2 formation |  |
| User creates their team from dropdown lists in a html table |  |

**User Inputs**

1. Register details (username, password, clubName, favteam, age, email, phonenumber)
2. Userteam (selected from dropdown list of footballers)

**Website Inputs, Processes and Outputs**

**Inputs**

* Size of browser window
* footballer , userteam, userdetails tables
* Login query results
* Registration query results

**Outputs**

* Error messages
* Login messages, and login status using session variables
* Registration messages – for successful/unsuccessful registration
* Search results for other user teams
* Finished user team with images and name on pitch
* How it works page

**Processes**

* Opening and closing connection to database
* Validation of user inputs
* Store user team and user details in database
* Populate html tables/drop down lists with data from database
* Utilise session variables to use personal user info across web pages
* Execution of SQL queries
* Place user team on pitch using multi dimensional array of positions

All inputs, outputs and processes mentioned in the analysis stage are also present or used in my final website.

**Evaluation**

**Fitness for Purpose**

After the conclusion of testing I would say that my project is fit for purpose as it meets the AH project requirements by integrating with three database tables to create a functioning website and utilises media queries to adjust to the screen size as well as matching all requirements stated in the analysis stage, both functional and end-user. I have produced evidence of my test case results, and they all display the expected result.

**Maintainability**

I believe that my solution is maintainable as I have used meaningful variable names throughout (PHP and HTML input elements) and have used comments on each page to explain the purpose of the section of code or line of code to allow other developers, should the website be taken to a larger scale by a team, to understand what is happening, which is evident in the code screenshots in the implementation stage. I have also used plenty of white space to allow others to clearly be able to read the code and distinguish what is happening at each line. However, on reflection, my code isn’t completely modular as I had the connection details and establishment of the connection to the database at the top of every page of code, when I could’ve used the php include statement and put these details in another file and then use the include statement (1 line) to retrieve this code, which I initially tried to implement but ran in to errors so opted to have them in every page to prioritize a working solution for the deadline. Also, if I were to increase the number of footballers in the football database I would need to find another way to store the footballer images, as eventually they may take up too much space on the computer hard drive and cause slow performance. Also, if I were to scale up the website so that more user’s would be using it I would have to change how the user selects other user names as the selectable dropdown list may become too long for the page.

**Robustness**

The registration and detail processes are robust as every single input on the register form is validated by the HTML form. Validation is also in place to find matching usernames on registration to ensure there are no duplications, and logins are only successful with the correct credentials. All of these validations will result in a message relayed to the user. Features elsewhere on the website are also robust as the user can’t view their own team when they don’t have one created (error message shown and user redirected to home page) , and can only select their team and the user their comparing their team with based on pre-existing options in a dropdown list populated by the database. If they then proceed to enter a team with duplicates then an error message is shown and the user is made to build their team again, without the duplicate entries. Therefore, I would say that my solution is robust.

There were a few features that I would’ve liked to develop but ran out of time and they ended up being unrealistic for the time frame. These features could perhaps be developed in the future –

* Admin login – option to add new players to footballer database table
* Most popular player message on homepage – count of the player that appears the most in userteam table, displayed on homepage
* Formation picker – allows the user to pick how their team shows on the pitch image, as well as changing the number of defenders/midfielders/attackers, (but total players in userteam still to equal 11) for example a 3-5-2.