**Name: Jasjeet Singh Siddhu**

**Registration No: 19BCE1423**

**Group members:19BCE1402(Aditya Agrawal)**

**19BCE1754(Shubh Pachauri)**

**Slot: G2**

**Topic: Football Simulator**

**(Premier league Simulator)**

**DBMS PROJECT REPORT**

**Introduction-**

A group of people posing for a photo

Description automatically generated

The Premier League is one of the most adored and hotly contested football leagues in the word patronized by an adoring fanbase around the world consisting of millions of people around the globe. This has led to the development of fantasy premier league game simulator by the Premier League on their website to increase the engagement of the Premier League and make them feel more involved. We decided to create a Premier League simulator to give the users a first-hand experience of the competitiveness of the matches played in the Premier League . This is a one of a kind simulator which aims to give users a realistic experience of the football action.

**Abstract-**

We have chosen to simulate matches between the top 10 teams of the Premier League because these teams have a bigger fanbase resulting in more users. To make use of the simulator the user has to first create an account in the simulator. The password of the user is hashed is hashed for security reasons. The user will be showed the updated lineups in accordance with the current football season. The user will then choose his/her team and the opponent. The match will be simulated and the result will be displayed according to the strength of the respective teams

This project report revolves around the various salient features of the project. It tells about the foundation of the project, why PHP is being used, use of mysql in the project, existing works like these, future of football simulators, architecture of the project, the algorithm used, a thorough run through of the website designed through screenshots and all the references which made this project possible.

Premier League Simulator-Present and Future

# Existing Simulators

**1)Football Manager**

A screenshot of a computer

Description automatically generated

The football Manager simulator has a wide database of players teams and leagues. It is a

Simulator which does not offer a 3D game engine. It works like a normal simulator and its design

is such that it gives a user which makes the user feel as if he/she is the real manager of the team

Its disadvantage is that it is not free.

**2)FIFA**

A picture containing street, sign

Description automatically generated

Fifa 20 offers a 3D game engine along with a simulator.This enhances the user experience but its main problem is its accessibility in that not everyone can afford a high tech gaming laptop and even if someone does have a gaming laptop, the cost of the game is high

**Future of our simulator**

We aim to add a 3D game engine to the simulation to enhance the playing experience and we would also like to add the remaining 10 teams to the database so that the game as close to reality as possible**.**

**Why Use PHP and phpMyAdmin ?**

**PHP-**



Python has a very clear advantage over PHP .However PHP is still relevant due to ease of use.

Compare to PHP Python has lower number of Frameworks. Popular ones are DJango, Flask. PHP has huge number of framework. Popular ones are Laravel, Slim.

PHP: Hypertext Pre-processors invented in the year of 1995 also known as PHP, it is a server-side scripting language. It is used for creating dynamic HTML content over the web. It is popularly used for generating XML documents, Flash animations, graphics, PDF files and much more.

PHP is an OOP language used only for web applications only its very robust and easy to use. So even a poorly coded web application will work that is not in case for python.

Its faster than python and cheaper to use .

**phpMyAdmin-**



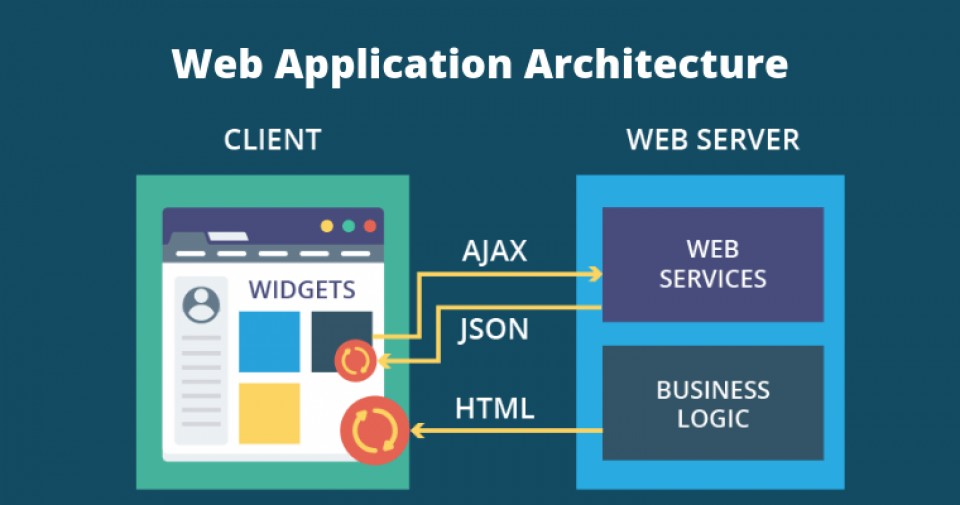
It’s a powerful tool created by PHP, used to handle MySQL over the web .It supports a wide variety of verities features on MySQL and Maria DB.

Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

Features

* Intuitive web interface
* Support for most MySQL features:
* browse and drop databases, tables, views, fields and indexes
* create, copy, drop, rename and alter databases, tables, fields and indexes
* maintenance server, databases and tables, with proposals on server configuration
* execute, edit and bookmark any SQL-statement, even batch-queries
* manage MySQL user accounts and privileges
* manage stored procedures and triggers
* Import data from CSV and SQL
* Export data to various formats: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, LATEX and others
* Administering multiple servers
* Creating graphics of your database layout in various formats
* Creating complex queries using Query-by-example (QBE)
* Searching globally in a database or a subset of it
* Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link and much more...

**Architecture of the Website (PROJECT)**

****

The project(website) is built with the help of –

HTML, CSS and PHP being used as the server-side language.

The database language used is MySQLi which is a OOP extension of the relational database management system (RDBMS) MySQL .

Though both MySQLi and MySQL accomplish the same thing, which is interacting with MySQL from PHP , MySQLi

Support some things that the old MySQL doesn’t . Things like

Prepared statements, multiple statements, and transactions are supported by MySQLi.

The database used in the project is phpMyAdmin

phpMyAdmin is a free and open source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services.

**Technological constraints-**

**1.Server- MAMP( Mac OS X, Apache, MySQL, PHP)**

Mamp is a free local server which is being used in the project.

It is a variation of LAMP software. It can be used to run a live web server from Mac, but is most commonly used for web development and local testig purposes.

Apache (or "Apache HTTP Server") is the component used to configure and run the [web server](https://techterms.com/definition/web_server). Once installed, Apache enables a Mac to host one or more [websites](https://techterms.com/definition/website). By configuring and running a local Apache web server, web developers can view their webpages in a web browser without publishing to an external server.

MAMP also includes MySQL and PHP. These 2 components are common technologies used for creating dynamic websites.

**2.Storage and memory**

As the website is meant to run on the client machine, all the pages are stored on the client machine.

It does not occupy more than 10 mb of space on the client machine.

**Functional Aspects of the website-**

The websites solely works as entertainment purpose and nothing more.

It is a little game for the football fans who can choose their and their opponents team and check out if their team has won the match.

Thus it shows the match results.

**Visual Aspects of the website-**

The website showcases various visual elements.

There are many buttons made with the help of CSS and basic HTML elements on all the web pages.

These button change their colour and cursor shape when the cursor hover over them.

There are interactive forms in the login and sign up page.

The football league table is also displayed with all the different colours for different teams.

Finally the Match page shows the images of users team and opponent teams along with Username of the user entered.

There are interactive links in all the pages.

There are different background images in all the pages.

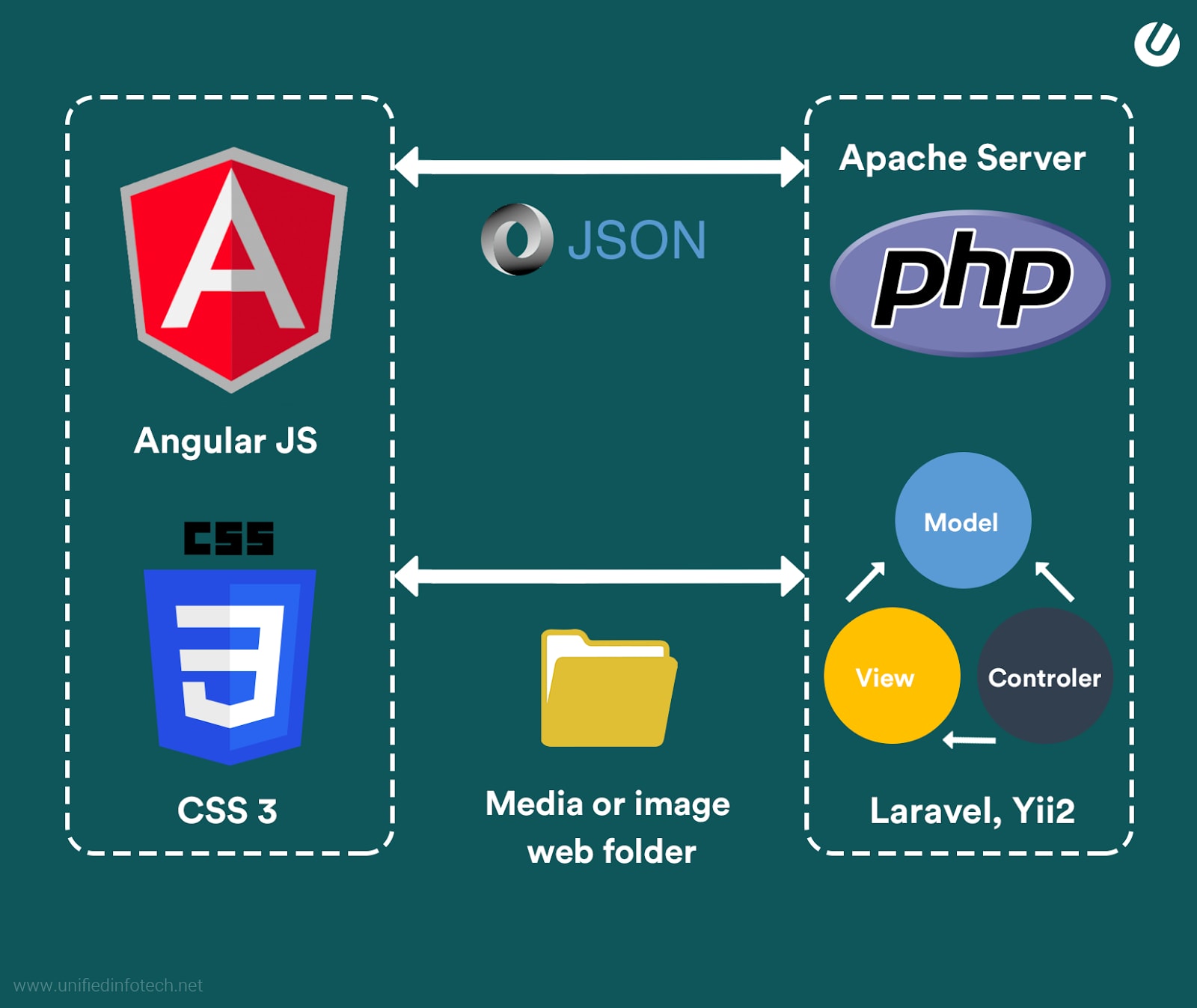
In some pages the background images are static and in some pages, the background images moves with the user control.

**Security Parameters-**

The website will guarantee total privacy of the user information and total protection as it does not let any duplicate user to enter and does allow any user with wrong password to enter.

The password given by the user during sign up is protected in the database as it is hashed before entering in the database.

This way the website is completely safe and offers complete protection.



The server side language used is PHP which receives the HTTP response from the user

**Algorithms-**

**When Local server is started with MAMP and Localhost is searched. It then directs to index.php file.**

db.php file-

1.Start

2.Define php variables DB\_SERVER storing localhost, DB\_USERNAME storing root , DB\_PASSWORD storing root, DB\_NAME storing database name accounts.

3.Create $link variable which connects to phpMyAdmin database with DB\_NAME and matches the credentials

DB\_SERVER, DB\_USERNAME, DB\_PASSWORD.

4.If the connection is successful , $link stores true or stores false

5.If $link==false the session dies and displays Could not connect.

A screenshot of a cell phone

Description automatically generated

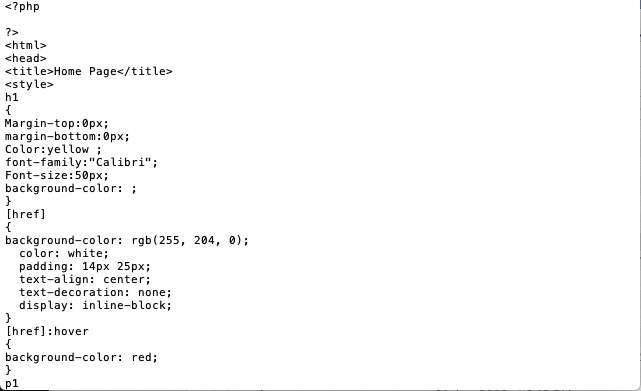
index.php file-

1.Start

2.Layout and structure by HTML and CSS.

3.You can press on the various buttons like home, contact, faq, teams list and you will be directed to the respective pages.

4.End



sign.php file-

1.Start

2.Layout and Structure with help of HTML and CSS

PHP part-

3. A connection is made to the database ‘accounts’ by using the command require\_oce “db.php”.

4.This calls the db.php file and a successful connection is made.

5.It checks whether the user has submitted the form. This REQUEST\_METHOD is known as POST.

6.If the form has been submitted , it checks if the username is not empty.

7.If username is empty then the user is redirected to the signup page and a error saying ‘emptyusername’ is displayed, the session dies.

8.If uername is not empty, a php variable $sql is made which prepares a select statement ="SELECT \* FROM accounts.users WHERE username = ?;"

9.Then the variable are bind to the prepared statement as parameters by using mysqli\_stmt\_bind\_param($stmt, "s", $param\_username);

10.Then the prepared statement is executed.

11.If successful , it is checked if the username already exists in the database.

12.If the username already exists, then the user is redirected to the signup page and the session dies.

13.If username does not exists then , then $username stores the username

14.Now we check the password entered

15.First we check if password is empty, if password is empty then the user is redirected to signup page with error.

16.If password is not empty, we check the length of password.

17.If length of password is<5 , the user is redirected to the signup page and error is displayed and the session dies.

18.If 17,16 is passed $password stores the hashed password

19.Now $name stores the entered name and $email stores the entered email by the user.

20.It is checked if both of them are non empty.

21.If either one is empty then the user is redirected to the signup page and the session dies.

22.If neither one is empty then a statement is made

$sql = "INSERT INTO accounts.users (username,name,email,pwduser) VALUES ('$username', '$name', '$email', '$password')";

23.This statements stores the name, user name, email id, password of the user in the users table

24.Another statement is made

$sql2="Insert into accounts.teams (username2) values('$username');";

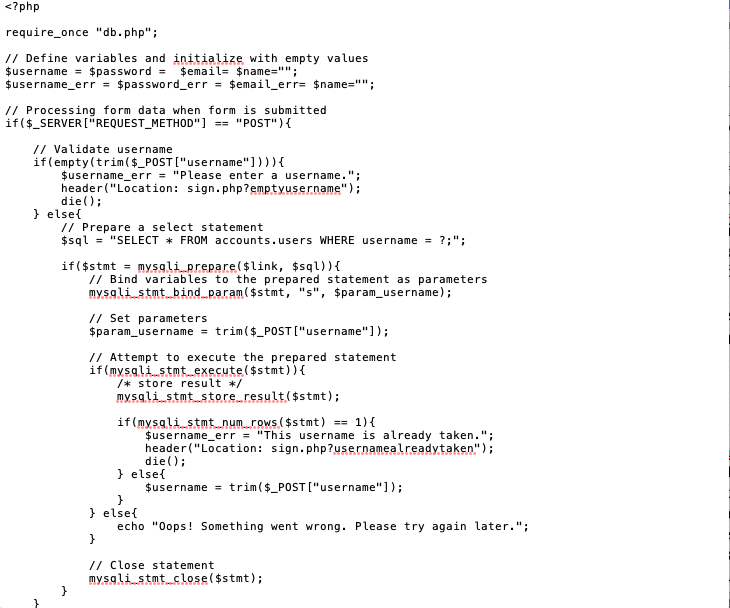
mysqli\_query($link,$sql);

25.This stores the username of person in teams table.

26.The connection is then closed for sign.php filr

27.end

**If the sign up is successful, the user is directed to the log in page , log.php file.**

****

log.php file-

1.Start

2.Layout and structure with the help of HTML and CSS

PHP part-

3.A session is started by session\_start()

4. First we check if the user is already logged in

5.If true the user is redirected to the welcome page and the the user exits the log.php file.

6.Connectuon is made to the database by using require\_once “db.php”

7.This calls the db.php file and a successful connection is made.

8.It checks whether the user has submitted the form. This REQUEST\_METHOD is known as POST.

7.If the form has been submitted , it checks if the username is not empty.

8.If username is empty then the user is redirected to the log in page and an error saying ‘emptyusername’ is displayed, the session dies.

9.If the username is not empty , then $username stores the username entered by the user

10.Then the password is checked whether it is empty or not.

11.If the password entered is empty, the user is redirected to the log in page and the session dies

12.If the password is not empty then $password stores the password

13.Now we validate the credentials

14.A select statement is prepared

$sql = "SELECT username, pwduser FROM users WHERE username = ?";

15.Now we bind variables to the prepared statement as parameters.

16.Now we execute the prepared statements and store the results

17.It is checked whether the username already exists.

18.If mysqli\_stmt\_num\_rows($stmt) == 1

Which means that 1 row exists with the given username,

We check the password.

19.password\_verify() is used to verify the entered password and the password stored in the database.

20.If true then the session is started and loggedin stores true and username stores the entered username.

21.The user is then directed to Team Selection page , TeamSelection.php file.

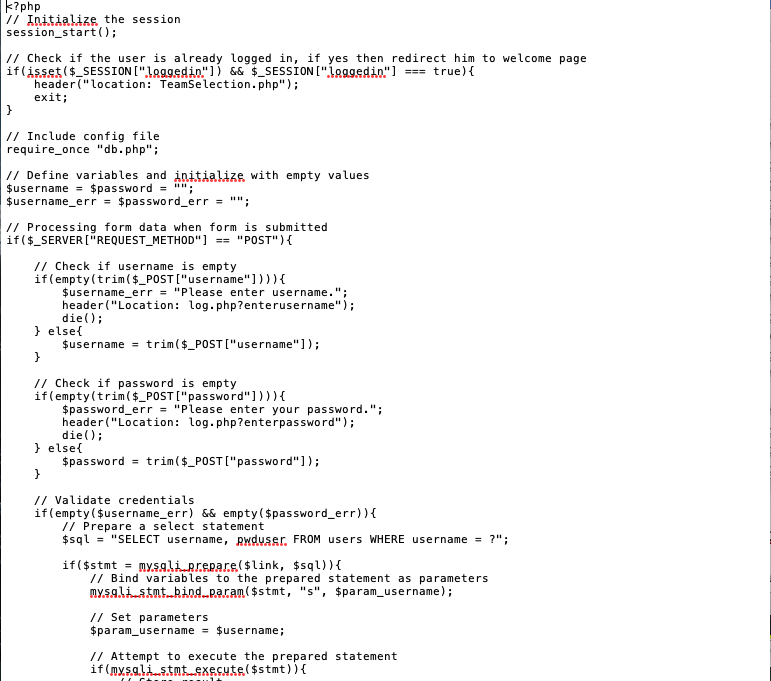
22.If 19 is false, the user is redirected to login page and the error is displayed

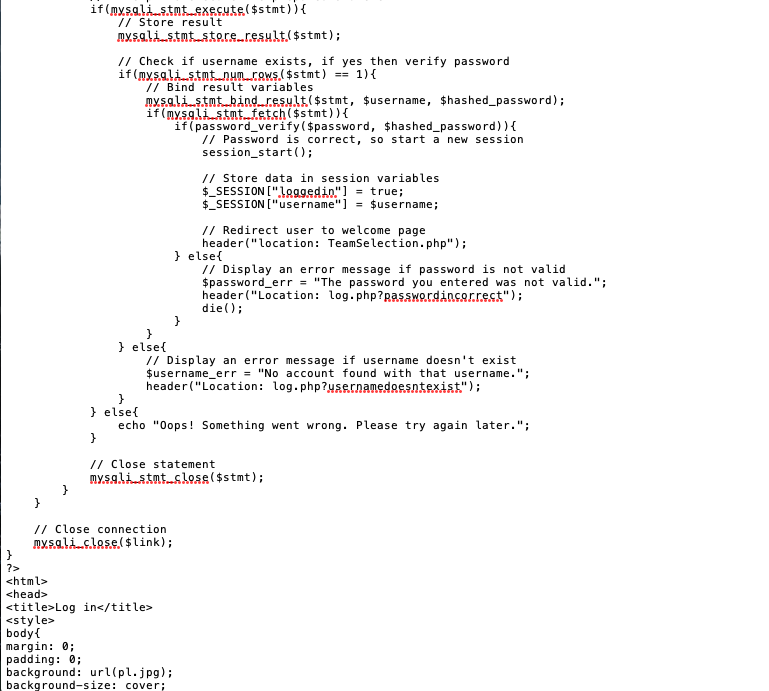
23.If 17 is false, the user is redirected to login page and the error is displayed

24.After the execution of the php file, the connection is closed with mysqli\_close($link);

25.End

**If the login is successful, the user is directed to Team Selection page, TeamSelection.php**

****

****

TeamSelection.php

1.Start

2.Layout and design of page is done with the help of HTML and CSS

PHP part-

3.A connection is made to the database ‘accounts’ by using the command require\_oce “db.php”.

4.This calls the db.php file and a successful connection is made.

5.It checks whether the user has submitted the form. This REQUEST\_METHOD is known as POST.

6.If the request method is post, it is checked if both the user and opponent has selected the same team.

7.If both have selected the same team, the user is redirected to the Team Selection page.

8.If not , the $user stores the name of the users team and $oppo stores the name of the opponents team

9.$\_SESSION[“team1”] stores the name of users team and

$\_SESSION[“team2”] stores the name of opponents team.

10.A sql statement is prepared

$sql="update teams set user\_team='$user', opponent\_team='$oppo' where username2='$se';";

This enters the users team and opponents team in the table ‘teams’ beside the username in the table.

11.Now the user is directed to the Match page , Match.php file and the control exits this file.

12.End

**After the teams have been successfully selected , the user is directed to Match.php file which displays the final score.**

**A screenshot of a social media post

Description automatically generated**

Match.php file-

1.Start

2.Layout and design of the page with the help of HTML and CSS

PHP part-

3. A connection is made to the database ‘accounts’ by using the command require\_oce “db.php”.

4.This calls the db.php file and a successful connection is made.

5.A session is started by session\_start()

6.$team1 stores the users team

7.$team 2 stores the opponents team

8.$sq1 stores the prepare statement to retrieve the path of image of users team.

9.$sq2 stores the prepare statement to retrieve the path of image of opponents team.

10.$sql3 stores the prepare statement to store the win probability of users team.

11. .$sql4 stores the prepare statement to store the win probability of opponents team

12.$query1 executes $sql1.

13.$query2 executes $sql2.

14.$query3 executes $sql3.

15.$query4executes$sql4.

16.$row1 stores the result obtained after executing $query1.

17.$row2 stores the result obtained after executing $query2.

18.$row3 stores the result obtained after executing $query3.

19.$row4 stores the result obtained after executing $query4.

20.Then it is checked which team has higher winning probability and a score is generated bases on there winning probability

21.The score between the teams is stored in s1 and s2 and later displayed on the page by echo() method.

22.The execution of Match.php is completed

23.End

**After Match.php is completed, the user has to sign out and logout.php file is executed.**

**A screenshot of a cell phone

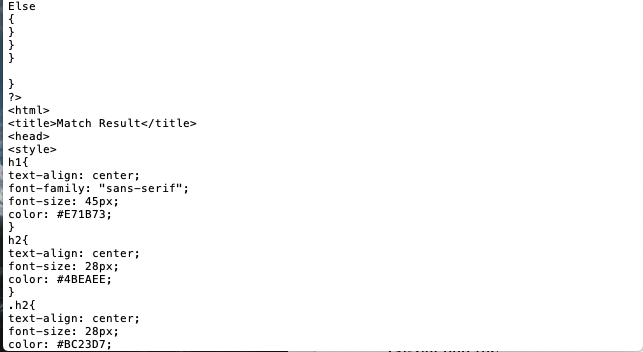
Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

****

Logout.php file-

1.Start

2.A session is started with session\_start()

3.All the $\_SESSION variables are made null by

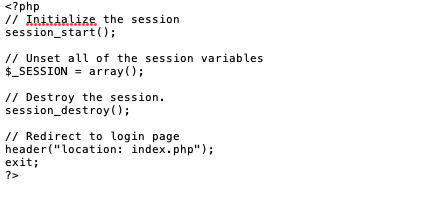
$\_SESSION = array();

4.$\_SESSION is then destroyed by session\_destroy();

5.The user is then directed to the index.php file and the control exits the logout.php file

6.End

**This way after login out the user comes back to the welcome page (index.php file) and can log in or sign up again and play again.**

****

**Screenshots**

1.Starting local server through MAMP

A screenshot of a cell phone

Description automatically generated

2.You are then redirected to MAMP site showing that the server has successfully been created

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

3.Then go to localhost page which redirects you to index.php file which is your home page.

A screenshot of a cell phone

Description automatically generated

4.Click on the sign up button and you will be redirected to the signup.php page. Enter your credentials in the form

A screenshot of a cell phone

Description automatically generated

5.If you have successfully entered the credentials then , your details will be reflected in phpMyAdmin database

Password will be hashed for security reasons.

A screenshot of a computer screen

Description automatically generated

6.You will be redirected to Log in page after sign up.

A screenshot of a cell phone

Description automatically generated

7.After successfully logging in you will be directed to the Team Selection page of the website. Choose your and yours opponent team.

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A close up of a logo

Description automatically generated

8.These choices are also reflected in the phpMyAdmin database.

A screenshot of a cell phone

Description automatically generated

9.Finally the score after full time is predicted after the algorithm is run and is displayed.A picture containing clock, dark, sitting, time

Description automatically generated

A sign in the dark

Description automatically generated

10.After playing you can sign out and you will be directed to the Home page again.

A screenshot of a cell phone

Description automatically generated

Screenshots of all the tables in the database **accounts-**

A screenshot of a computer screen

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

**Screenshots all the pages in the home page-**

Teams List-

A screenshot of a cell phone

Description automatically generated

Contact-

A screenshot of a video game

Description automatically generated

FAQ-

A picture containing red

Description automatically generated

Rules-

When Rules button is clicked, a word document containing all the rules is downloadedA screenshot of a cell phone

Description automatically generated

Sign up-

A screenshot of a cell phone

Description automatically generated

Log in-

A screenshot of a cell phone

Description automatically generated

**References**

1.  [HTML Full Course - Build a Website Tutorial - YouTube](https://www.youtube.com/watch?v=pQN-pnXPaVg)

2.  [Learn HTML5 and CSS3 From Scratch - Full Course - YouTube](https://www.youtube.com/watch?v=mU6anWqZJcc)

3.

<https://www.w3schools.com/html/>

4.

<https://www.w3schools.com/css/>

5.

<https://htmlcolorcodes.com/resources/>

6.  [How to Create Registration Form in HTML - Easy Step - YouTube](https://www.youtube.com/watch?v=D4jj3HHrnSU)

7.

<https://www.youtube.com/watch?v=WZ2uqGkHoR0&list=PLLAZ4kZ9dFpNO7ScZFr-WTmtcBY3AN1M7>

8.

<https://www.youtube.com/watch?v=B6CoFlIgh1k&list=PLLAZ4kZ9dFpNJvT-JKogcCySuSznPpXEp>

9.

https://tutorialvilla.com/how/how-to-run-first-php-program-in-xampp-server

10.

<https://editrocket.com/articles/php_mac_os_x.html>

11.

<https://stackoverflow.com/questions/20105029/include-php-file-into-html-file>

12.

https://www.php.net/manual/en/function.header.php

13.

<https://www.w3schools.com/php/>

14.

<https://stackoverflow.com/questions/16616174/displaying-username-using-sessionusername>

15.

stackoverflow.com

16.

<https://stackoverflow.com/questions/30600654/insert-image-datatype-in-sql-server-table-using-string-concatenation>

17

<https://stackoverflow.com/questions/52265954/the-requested-url-learnphp2-upload-php-was-not-found-on-this-server>

18.

<https://stackoverflow.com/questions/19323010/execute-php-function-with-onclick>

19.

<https://www.peachpit.com/articles/article.aspx?p=1315026&seqNum=6>

20.

<https://www.google.co.in/?client=safari&channel=iphone_bm>

21.

<https://www.quora.com>