GUM

ReadMe File

****
****

What is our database about?

We selected **Field Hockey** as our sport of choice.

Our Field Hockey Tournament is not gender-specific, but rather **gender-inclusive**, so all genders may participate.

This database represents what is going to happen in the sport, i.e., the **Tournament**.

As well as managing general users and administrators who will login to the websites, it contains the details of what could happen during the tournament.

Users interface is optimised and handled by **PHP**, which is communicated with by **phpMyAdmin**. The purpose of this is to assist with query creation, table management, and using **C.R.U.D.** rules.

****
****

What works in the database vs. what doesn't?

- **Works** -> Everything works.
- **Doesn't work** -> **Our Assumption** : Tournaments cannot be deleted because the concept and idea of a database is to allow users to view and see records of data that happened years ago.

  Deleting a tournament causes many constraints and makes the integrity and data validation of the database 'loose', meaning information/data coming in and out of the database is neither validated nor coherent.

  Also by following real world laws, when a record has been created and/or not completed due to some unforeseen reasons, the record of that must be kept **by law of the Data Retention Policy.**

Works	Does not work
Login and management of users	Cannot delete tournaments
Management of teams, players and organisations	
Capturing of scores for tournament, the events and players	
Uploading of media for the tournaments	
Produce of statistics for a tournament	

****
****

Our Research?

Refer to the GUM Practical 5.pdf in the Research section.
https://github.com/Marumo-ProG/GUM-Hockey-App-COS-221-/blob/ccc709db6c06431e425022602de85f9723669066/GUM%20PRACTICAL%205.pdf

****
****

Our Data Validation

Front-end Validation:

For **password validation**, REGEX was used:

- A password must be longer than 8 characters
- A password must capital letters, small letters as well as symbols

Reason : useful in terms of search and replace operations.

For **email address validation**, REGEX was used:

- An email address must contain the @ symbol

Reason : useful in terms of search and replace operations.

```
function
validateform(){
                        var name=document.myform.name.value;
                        var password=document.myform.password.value;

                        if (email==null || email==""){
                         alert("Email can't be blank");
                         return false;
                        }else if(password.length<6){
                         alert("Password must be at least 6 characters
                        long.");
                         return false;
                         }
```

SOURCE:
https://github.com/Marumo-ProG/GUM-Hockey-App-COS-221-/blob/main/login.html

Back-end Validation:

For **password validation**, Blowfish hashing was used.
Reason :Easier and fast encryption and security of data. Harder to decode/hack/decrypt.

For **email address validation**, using SESSIONS.
Reason : It lets the server know that all requests made, i.e. login in, POST, UPDATE, GET Requests originate from the same user, thus allowing the site to display user-specific information and preferences.

```
$email =
$_POST["email"];
                        $password =
                        $_POST["password"];



// hashing the password before storing
it in the database;
                                        $password = password_hash($password,
                                        PASSWORD_DEFAULT);
```

SOURCE :
https://github.com/Marumo-ProG/GUM-Hockey-App-COS-221-/blob/main/validate-signup.php

****
****

Usage of package manager

No package managers were used i.e. NodeJS, Maven.
PHP was used throughout the development of the website.

****
****

Our git commit messages

Proof of our git commit messages

Add the commits page website here:
Refer to the GUM Practical 5.pdf in the Git Commit and Command Section.

https://github.com/Marumo-ProG/GUM-Hockey-App-COS-221-/blob/ccc709db6c06431e425022602de85f9723669066/GUM%20PRACTICAL%205.pdf

https://github.com/Marumo-ProG/GUM-Hockey-App-COS-221-/graphs/contributors
****
****

Uploaded PDF

Commit the GUM Practical 5.pdf to Github.

Add the link here:

https://github.com/Marumo-ProG/GUM-Hockey-App-COS-221-/blob/ccc709db6c06431e425022602de85f9723669066/GUM%20PRACTICAL%205.pdf
****
****

Command Lines when using a Command Prompt

This section is under the Development page in the Field Hockey.pdf as proof.

The database can be displayed in MySQL Client (MariaDB 10.5 (x64)) through the following example execution:

- SHOW DATABASES;
- SELECT * from users;
- SHOW ALL TABLES : SHOW TABLES;

  ```
  Adding a column with an unique, auto-incrementing ID: ALTER TABLE [table] ADD COLUMN [column] int NOT NULL AUTO_INCREMENT PRIMARY KEY;
  ```

****

How does our website work?

When using the website there are two types of users:

- **General users**
- **Admin users**

General users can view the vague interpretation of data from the database, this being the Overview page, overall stats and upcoming matches page without having to login. If a user wishes to login, the link "Manage Teams & Players" can be found in the footer.

Once a user is successfully logged in (a user can either be an **Admin user** or not), they would be redirected to the Dashboard page where they can view a more detailed description of data, manage teams, players, tournaments.

If a user would like to view the in depth statistics of the sportsDB, they would just need to click the "Stats" link where they would be redirected to the information we have on a player(offensive and defensive), coach and team. This information includes, the respective rankings, and performance statistics which reflects the relative entities ability and skill. For example, an offensive player would have their Position,Cards Obtained, Total Games, How often they have Started, Shot accuracy, Total Assists as well as their Name displayed in the "Offensive Player Stats". The same would be done for the defensive player, coach and teams entities with information that applied to them.

A user records the events that are taking place in a particular game through the DashBoard page(once they click the 'Record Games' link). The link redirects the user to a form where they will submit a form requiring details of the game such as the game's ID, Tournaments name, and the user's username. Once this form has been successfully completed. They will be redirected to an html page where they will be required to record each event that takes place in the said game.
These events being fouls, goals, shots, substitutions, corners, and the game winner. An event is required to be recorded with details that would be used to populate the database, with information required for the overall sportsDB.

For the Admin User

Default login credentials are needed:

```
##################################################
#     First Name: Is_admin                       #
#     Last Name: User                            #
#     Email: IamAdmin@gmail.com                  #
#     Password:1234@Is_Admin                     #
#     Admin Code: 12345                          #
##################################################
```

- An **Admin user** would be directed to a user management page, where they would manage the users of the website (delete and/or add).
- Within the **Dashboard page** it would display the currently ongoing Tournaments.

- By navigating to **Game Details**, it would display the ongoing/currently playing games, displaying their times, scores between each team, the umpire responsible for that specific game and the location the game is taking place in.
- In **Stats**, it displays the statistics of all players (both defensive and offensive), as well the statistics for their teams and their individual coaches.
- In **Players & Teams**, it enables a user to input the team's , Coaches as well as update tournaments information i.e for the Teams the name of the team can be inserted, their coaches id, their captain as well as their origin.
- In **Tournament management**, it enables a user to create their own tournament i.e giving the tournament a name, which season it's going to be played in and the country and city the tournament will be taking place in.
- In **Upload Media**, provides the user to upload various media files that were taken during the tournament , these could be images and/or files.

What works on the website vs. what doesn't?

- **Works** -> Everything works.
- **Doesn't work** -> **Our Assumption** : Updating the **Admin Users** information. This doesn't work nor is it implemented because updating information about the admin is crucial. This changes the entire website and database in the backend as many data validation rules will be violated.

  For example, if an admin user wants to update their email address or name, it will affect the database as the ID is linked/ validated with the email address which will need to be updated in the backend part of the website and validation that were done previously will need to be applied as well in the newly updated admin information.

****

Creators/Authors:

1. Tlhalefo Dikolomela - (u21507792)
2. Carey Mokou - (u21631532)
3. Thabang Kgaladi - (u21686875)

4. Emilee Da Cruz - (u21494577)
5. Hawa Ibrahim - (u20418494)
6. Lenny Thobejane - (u20485001)