**Year 13 Assessment Template – Creating a database-driven website – AS91902/3**

**Purpose - ChallengeMe**

* + This website will allow users to log on and become a part of any competition. They will be able to create an account and challenge other users. They can submit results, view the leaderboard, bet/vote on up and coming event, view the up and coming events, view other users and their account details along with adding, editing and changing basic account information. They should also be able to write to a forum or comment section. All users private information (passwords etc.) will be kept safe and secure from other accounts. An admin account should be able to view all pending events, view all pages, and create challenges between other users. Admins should be able to create new admin accounts. No page should be accessible from any unregistered or logged out account.

End-user requirements

* Users should be able to view all users rank on the leader board, view all up and coming events, challenge other users, and delete an up and coming (pending) event which they're affected by. User should also be able to comment on a forum or comment section to other users. Users can log onto their account or create a new one. Logged on as an admin will allow the user to delete any up and coming event, user, or challenge. The admin page will have its own admin panel which is more efficient for the admin to complete these tasks.

Stakeholders

* I will consider the opinions of a range of stakeholders for this site. These will mainly be the students (as the students are the ones using the site) so the likes of Jordan, Alan, Hugh, Harry etc. All of which would use the site if it went live.

## Stage 1 – Relevant implications and conventions

You may use an implication that you used in the design section. Remember that you must cover both the website and database when discussing your relevant implications. You must identify 2 relevant implications for your website (you have probably done this already in your design section) and 2 for your database - **they can be the same two implications if this is appropriate.**

Relevant implication 1: Identify a relevant implication that you need to consider. Describe what it is and explain why it is important that your final website meets this implication. Relate this back to your purpose/end user requirements

**Functionality (website)** - This refers to providing the user with basic factions which allow them to carry out tasks and processes that relate to the purpose of the website. Linking to pages, nav bars, requesting users to challenge, submitting results, betting, etc. will all appear accurately and work as expected. Error catching will be used to prevent basic user errors and provide them with feedback regarding the particular error. The website will be relatively responsive, allowing users on mobile or PC to view the same website in a functionable and eye-catching way. This is all required to produce a functioning website.

It is important for the purpose of the site that this implication is met. Being an information-based website, it is important users are able to log on, challenge others, view results, or bet on an up and coming event.

This comes down to functionable the website is. If the navbar doesn't work accurately, and/or links to user profiles aren't accurate and working then the user is simply unable to use the website to its full potential (it's purpose). Users will be unable to access certain key parts of the site and this fails to meet the purpose of the website.

If the betting system doesn’t work, or has errors in it, again this will result in a bad user experience, and the website not functioning properly. The results will be inaccurate, or non-existent, meaning the user won’t be able to use the website for its intended purpose properly.

Likewise, being accessible on a range of devices is important as users may wish to check this information on the site regularly or update personal information when without a PC. This means it must resize effectively, with a readable and useable display on all screen sizes. Failure to do this will lead to functionality issues as buttons may appear in the wrong order, shape or color.

Overall, the functionality of my website is extremely important as will ultimately determine the users experience and ability to fulfil the websites purpose.

Relevant implication 2: Identify a relevant implication that you need to consider. Describe what it is and explain why it is important that your final website meets this implication. Relate this back to your purpose/end user requirements

**Usability** **(website)-** This implication generally refers to the user experience within the platform, meaning organised, readable, accurate, and constant information is being displayed on the website. This may look like commonly accepted conventions (footer, navbar), accurate error messages, links when required etc.

I'm achieving these things and addressing this implication by maintaining a constant 'feel' across my website. Being a very information-based website, I will utilise white space, simple design, consistency is position and colour to create a common environment throughout the site making it easier for users to navigate around the platform. This will ultimately better the user experience. I will additionally provide features, such as ‘back’ button, with the aim of improving how efficiently users can navigate around the site. This, along with dates on events, admin quick links, leader board users which links to their profile, etc. will all help to improve the user experience. I will also have error messages that alert the user when a mistake has been made (e.g. password or username is incorrect).

Achieving this implication is very important for the platform, as it should remain simple, yet information based, allowing users to log on, view previous results, and challenge other users. If the website is hard to navigate users are less likely to log on and challenge others, bet on events etc. which may be a result of poor linking, consistency, or other design variations. Lack of error messaging may result in less user activity, as they simply aren't aware of the error that's occurred (e.g. have scheduled two challenges at the same time). So overall it becomes very important that these features are met, and the implication is achieved.

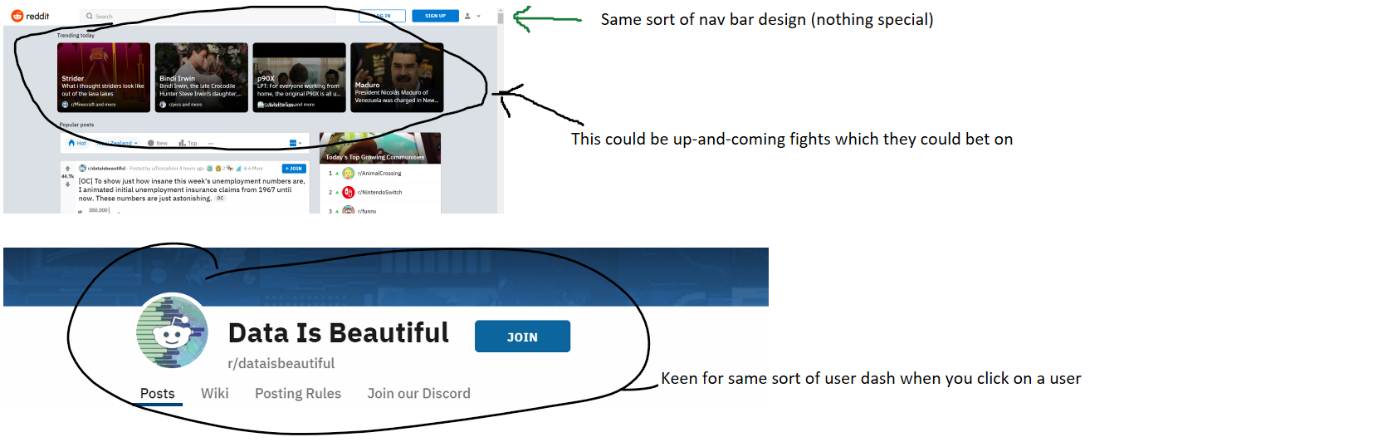
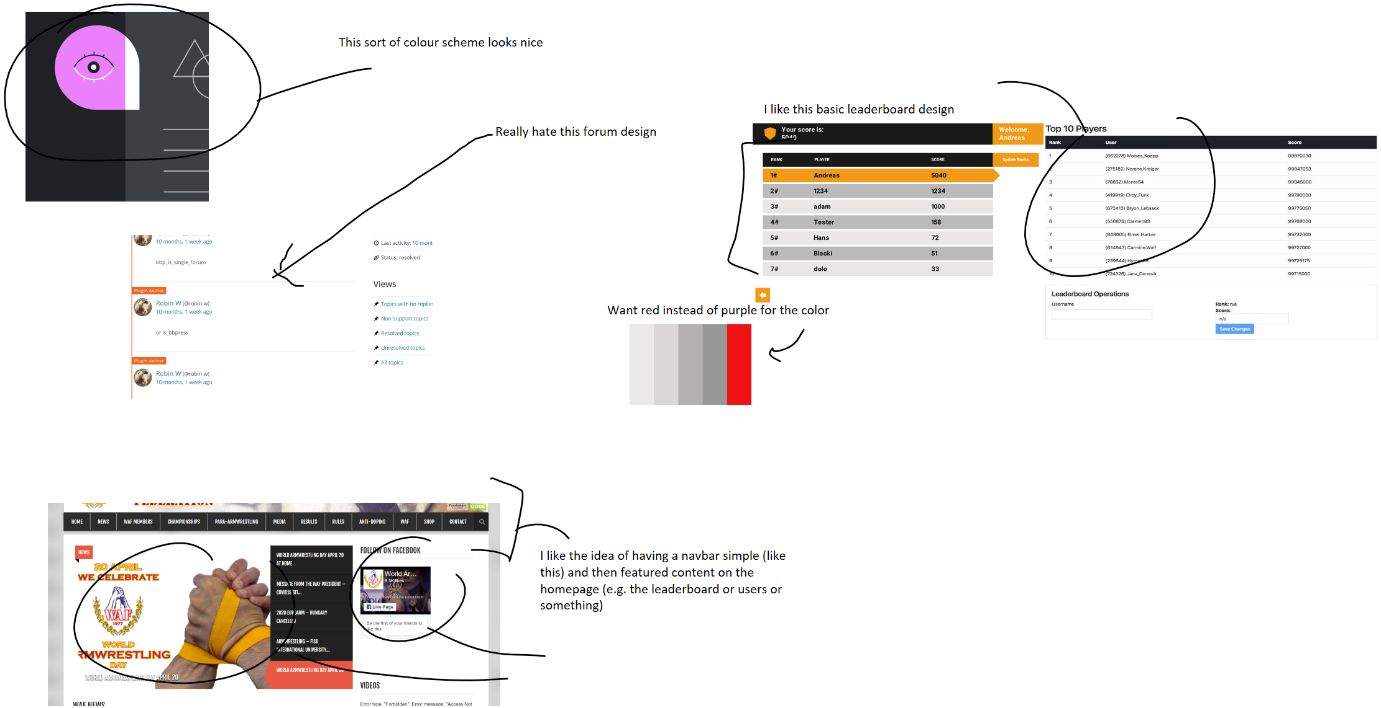
## Stage 2 – Establishing design of site and database and relevant implications

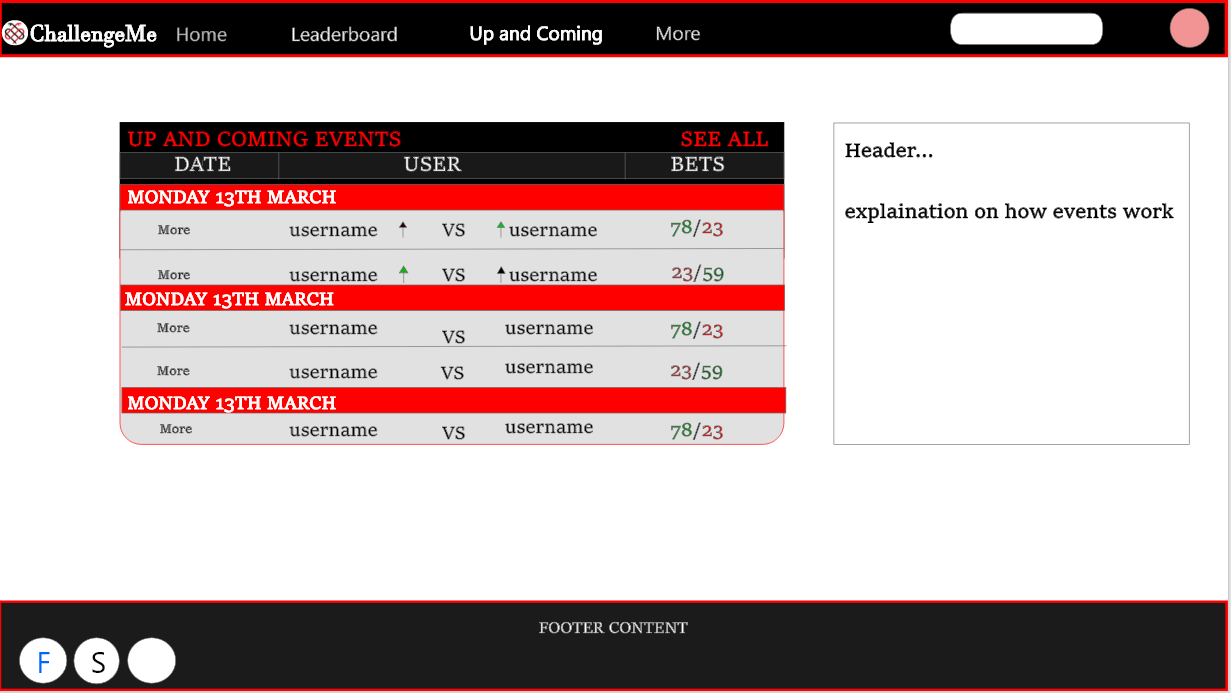
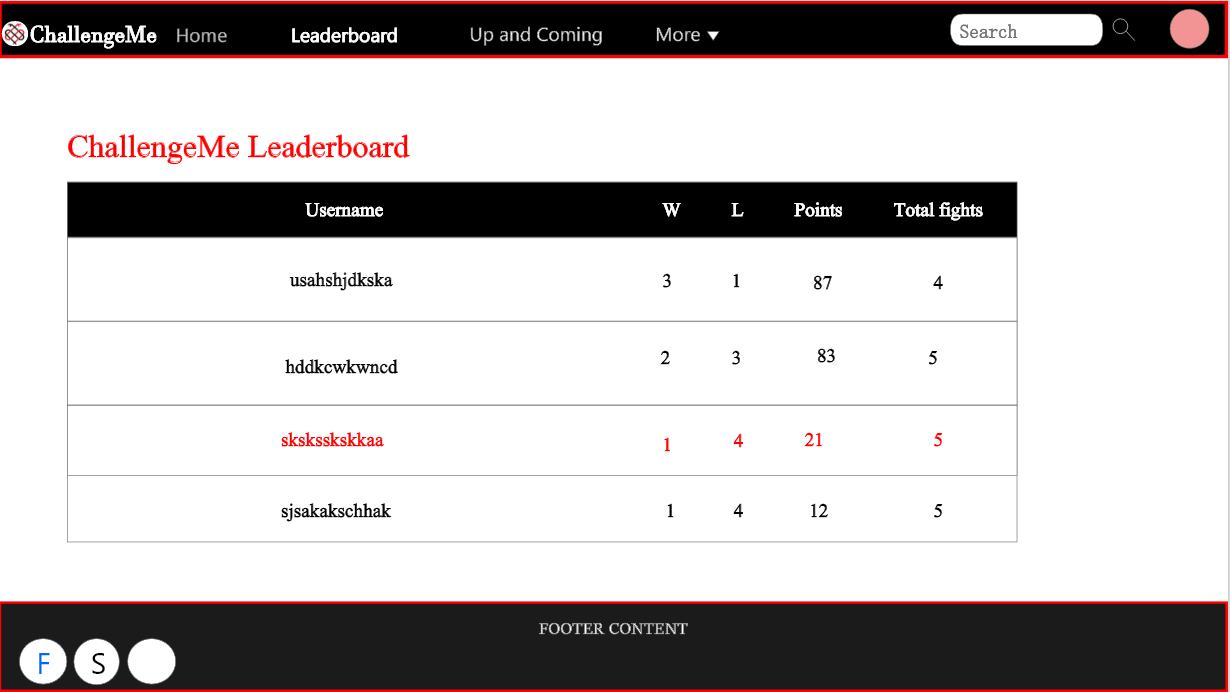
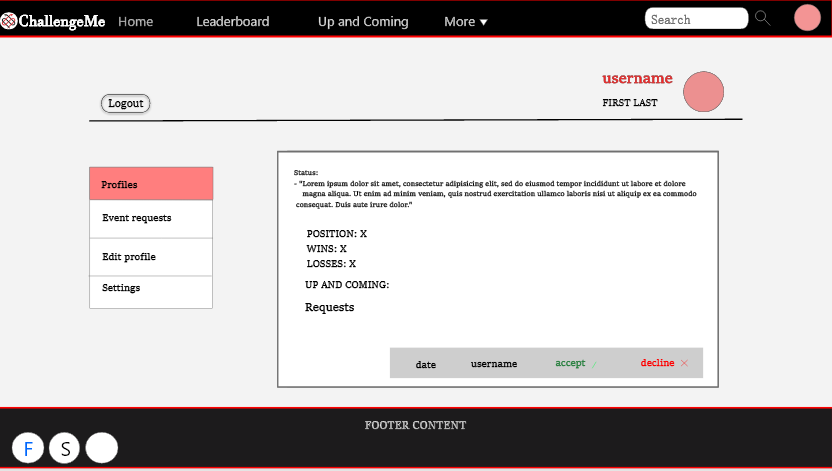
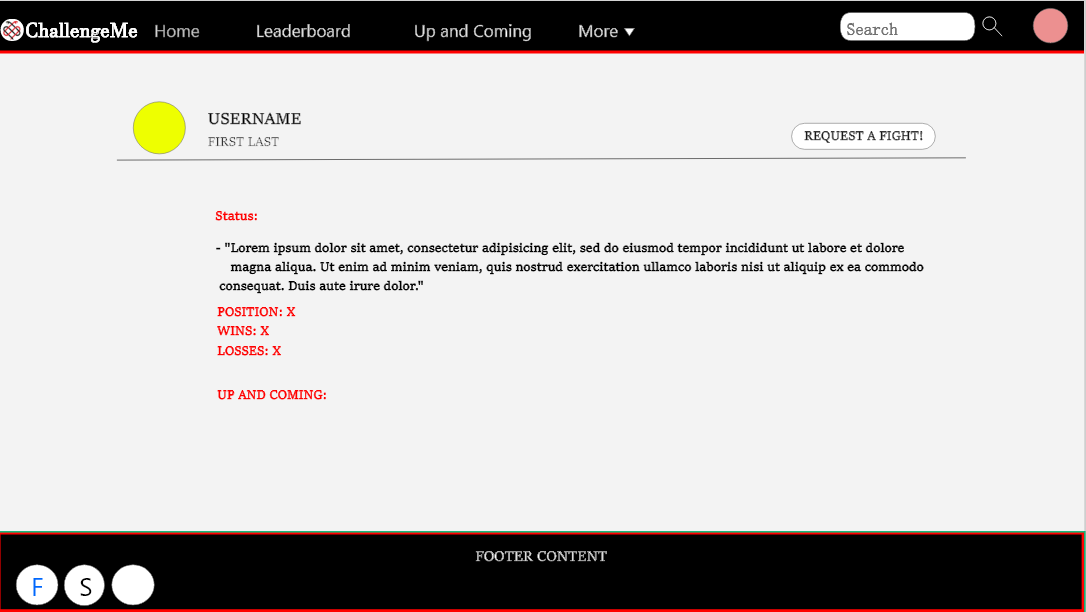
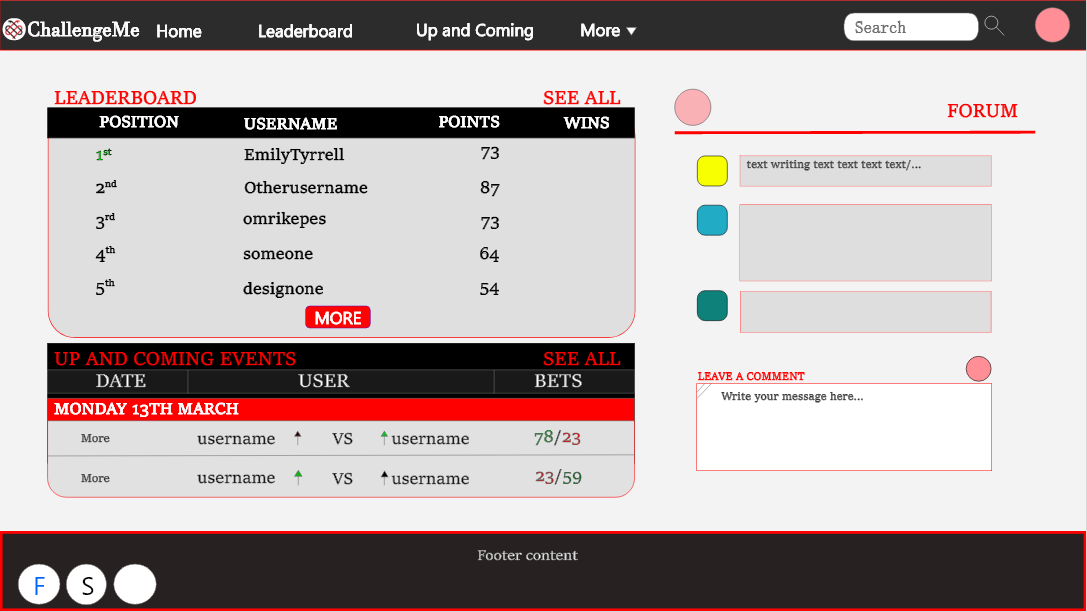
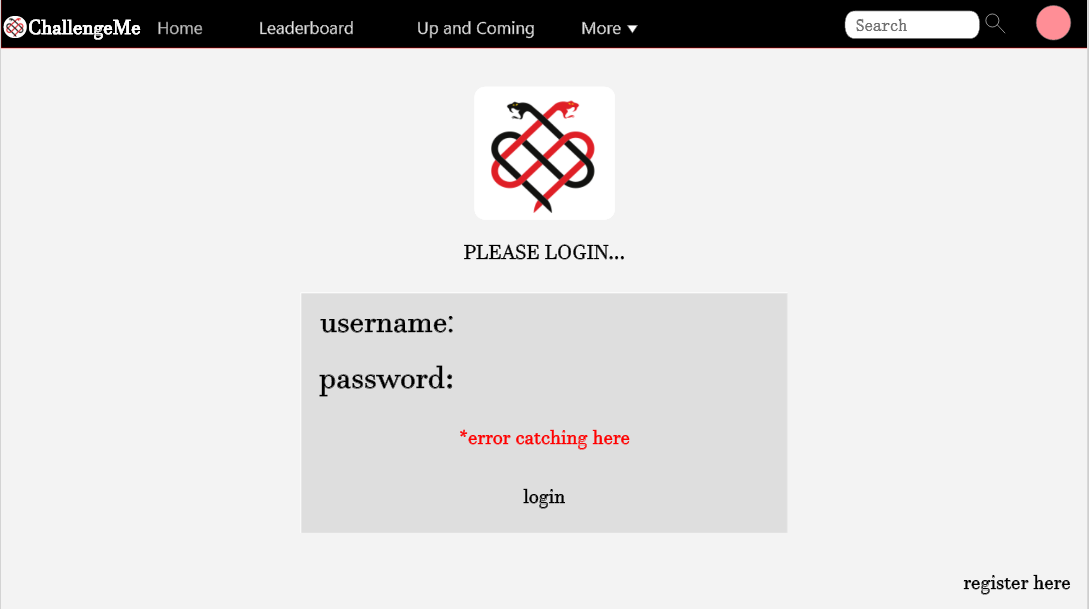
Site design: Include a screenshot here of the final design you came up with in the design standard. (It is okay if your design changes during the creation stage.)

* Early stage getting ideas.

Friday, March 27, 2020

11:39 AM

* 
* 



## Stage 1 – Relevant implications and conventions

You may use an implication that you used in the design section. Remember that you must cover both the website and database when discussing your relevant implications. You must identify 2 relevant implications for your website (you have probably done this already in your design section) and 2 for your database - **they can be the same two implications if this is appropriate.**

Relevant implication 1: Identify a relevant implication that you need to consider. Describe what it is and explain why it is important that your final website meets this implication. Relate this back to your purpose/end user requirements

**Privacy (database)** - The implication of privacy is based around keeping user data confidential and secure.

This may come in the form of limiting user accessibility to certain information (e.g. personal data). Ensuring personal property (images, names, birthdays etc.) aren't shared without permission is important and if this isn't respected, cause led to legal action.

My website will address this through a few key processes.

* User data, for example, usernames, first & last names, passwords etc. are stored in a secure database (password protected). The extremely sensitive information (user passwords) will be hashed (encrypted) within this database, should the database be hacked into, meaning even then, passwords become much harder to steal. This greatly increases the security of the website and user experience.

It's very important my website can meet these implications as providing a safe and secure platform is essential if users are entering private information. User experience will be determined by factors including how secure they feel their information is being handled.

.

Relevant implication 2: Identify a relevant implication that you need to consider. Describe what it is and explain why it is important that your final website meets this implication. Relate this back to your purpose/end user requirements

**Future proofing (database)** - This implication means assuring that the website/database that is being produced, is scalable and able to be extended upon in the future. This is very important as a lack of addressing this can result in poor data storage, inability to update certain aspects within the website, and decreased functionality within the website. My website will be addressing this by assigning each user a userID to help avoid any possible name duplication errors, and having these userID's automatically increase. This allows users to be added into the database quickly and simply and easily recognized within the code. Designing the database in such a way that it is the most efficient arrangement to increase the speed and amount of data being stored. It is very important this is achieved as failure to do so may result in website functionality decreasing, and thus the user experience and purpose of the website starts fail as well. If this website becomes live it is important the database is designed and created in such a way to handle potential variations in user behavior, duration, and number.

Database design (planning): Fill in the tables below with the names of the tables, column names, data types and lengths.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name:** | user | | |
| **Column names** | **Data type** | **Length** | **Comment(s)** |
| userID | int | 5 | Primary – auto increments |
| Username | varchar | 20 |  |
| password | text | 400 | Encrypted |
| firstName | varchar | 15 |  |
| lastName | varchar | 15 |  |
| userWins | Int | 6 | Default - 0 |
| userPoints | Int | 6 | Default - 0 |
| TotalChallenges | Int | 8 | Default - 0 |
| Level | Int | 1 | Default - 0 |
| status | Varchar | 400 | Default – “This user hasn’t added a status” |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name:** | Challenges | | |
| **Column names** | **Data type** | **Length** | **Comment(s)** |
| userOneID | Int | 5 |  |
| userTwoID | Int | 5 |  |
| winnerID | Int | 5 | Default - NULL |
| challengeID | Int | 6 | Primary – auto increments |
| date | Date |  |  |
| creatorID | Int | 5 |  |
| Status | Int | 1 | Default - 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name:** | Bets | | |
| **Column names** | **Data type** | **Length** | **Comment(s)** |
| challengeID | Int | 5 |  |
| betID | Int | 20 | Primary – auto increments |
| pickedID | Int | 5 |  |
| winnerID | Int | 5 |  |
| userID | Int | 5 |  |

Justification of database design: Explain why you have designed the database this way. i.e. Justify why you have used common keys across tables.

Every table has an ID column so it’s to be referenced by other tables and the website. For example, a challenge’s ID is put through a ‘get array’ so the website ‘knows’ what challenge to display on the ‘UpAndComing.php’ page.

**User:**

The length of ID for users is 5, which allows for about 100,000 accounts to be made. This seems high, but it means if this site went ‘big’ (similar to Amazon) then the maximum number of users will need to be increased progressively increased. There is also a column for ‘username’, ‘firstName’, ‘lastName’ and ‘password’. This is to allow users to login; these are all varchars so that text can be inputted. The username has a max length of 20, assuring usernames can’t be ‘to long’ as to affect the websites display. This is the same reasoning behind the 15-max length for user’s personal names.

Password has a length of 400 because passwords are encrypted when stored to add extra security and help with privacy. The length of 400 allows for a range of different encryption lengths like 64, 128, 200 etc.

UserWins stores the number of wins the user has. This is an Int with length 6 (e.g. 1,000,000 max) which is an optimistic view, yet not unreasonable, for the max amount of wins a user can have. This is the same for UserPoints, and for the same reasons. Users can can wins, and points, easily, meaning it’s important there is enough ‘room’ to store this information. Both of these have a default value of 0. This is because when first created, a user will have 0 wins and 0 points. This allows them to be displayed on the leader board.

TotalChallenges stores the number of events each user has competed in. This is also stored in an Int (number) and has a length of 8. It must be larger than the userWins, as not all events competed in, will result in a win. This also has a default value of 0 as users till automatically have 0 total challenges. This helps with percentage calculation and rankings of users.

There is also the user level, which determines weather a user is an admin or user. This is stored as an Int of 1 length. This isn’t a boolean as it allows for feature addition of levels of users (e.g. editor, admin, user, creator etc.). Users default will be 0 as they (if not specifically created as an admin) , are only allowed user permissions.

Status stores each user’s individual status. This has a max length of 400. This is meant to be small, as users’ statuses shouldn’t be long at all. Storing the data in a voucher allows for the text to be inputted. Users will have a set default when they a created. This allows some text to be displayed on their user profile, despite them not adding any before. This default accurately represents the user’s situation.

**Challenges:**

The length of ID for challenges is 6, which again, allows for about 1,000,000 different events. This is not unreasonable if the site were to expand to other schools, as any user can create a new event. There are columns for the winner, userOne and userTwo ID’s, each with a reasonable length of 5. The winnerID is unique as it has a default value of NULL, meaning when first created, a challenge doesn’t have a set winnerID. This allows the winner to be added later on and edited.

The column for date stored in the date format. This allows for the events to be added with a corresponding date, helping to remind users when each event should occur. This must be stored in the date format. CreatorID is the userID of whoever created the event, this allows for the request to then be sent to the other user involved. This is stored as an Int with a length of 5.

Lastly, status. Each event (challenge) created, has a status. This determines the stage at which the challenge is in. 0 representing a new challenge which hasn’t yet been accepted. 1 meaning the challenge has been accepted and is currently active (displayed on the up and coming page). 2 representing a challenge which has been completed, and a winner has been entered. And 3 meaning the challenge has been approved, and finished, and the users involved have had their userWins and points updated. This is stored in an Int of length 1, allowing for future introduction of other stages etc. This has a default value of 0, meaning all events created are automatically put into the ‘pending’ stage.

**Bets**:

The length of the ID for categories is 3, which allows for up to 1000 different categories. This is optimistic, but it does mean many different categories of book can be added which provides flexibility if the site becomes popular. The only other column, currently, is the name of the category which has a length of 30, suitable for a category name.

These common keys drastically improve efficiency as otherwise products added to a user’s cart would have to be stored inside the user table, which would result in a new column being needed for every product. This table is required as it means that the information of what a user has added to their cart can be store and accessed across multiple sessions. This table also has column for quantity that stores the number of the product the user has added to their cart. This allows for users to order multiple of an item efficiently instead of individually adding many of the same product to their cart. This column has a length of 2 to allow for a max quantity of 100 – a reasonable maximum.

Trello

Wednesday, April 1, 2020

4:37 PM

Machine generated alternative text:
Boards 
Website tas 
Things To Do 
ebsite 
6 
Private 
OK 
p 
Talk to stakeholders bout designs 
Edit design and work to solidify a 
design 
HTML basic design (not much 
function) 
Design databass 
Forum (commenting and betting on 
fi ghts) 
Leaderboard sections 
Add users (admin, user) 
Front end done 
+ Add another card 
Doing 
Research other websites for design 
deas 
First designs (XD, paper etc) 
Set of basic requirements from end 
user 
+ Add another card 
vit 
Done 
Solidify website idea (what it is, what it 
will be able to do) 
+ Add another card 

Start of April - working towards coding stages

Machine generated alternative text:
oards 
ChallengeM 
Things To Do 
Editing account details 
Account photos 
Challenging users to fight 
+ Add another card 
ebsite 
6 
Doing 
Private 
OK 
Done 
Solidify website idea (what it is, what it 
will be able to do) 
Research other websites for design 
deas 
Find the websites colour scheme 
First designs (XD, paper etc) 
Set of basic requirements from end 
user 
Talk to stakeholders bout designs 
Edit design and work to solidify a 
design 
+ Add another card 
another list 
Design databass 
Add users (admin, user) 
Leaderboard sections 
Basic HTML front end design complete 
(no real function) 
+ Add another card Machine generated alternative text:
ChallengeM 
Things To Do 
ebsite pr 
6 
Doing 
Private 
OK 
Basic leaderboard and up and 
coming pages functioning (talking to 
database and displaying results in 
order) 
Creating a new challenge page 
functioning (inserting a new 
challenge into the database) 
Upone selecting a user, displays user 
infomation (username, first name, 
wins, etc.) 
Home page (with mini leaderboard 
and mini up and coming) complete 
and funcational 
Creating sessions, i.e. Login page, 
and restricting the ability of certain 
users (intro of user level - admins, 
users etc.) 
Design databass 
Add users (admin, user) 
Leaderboard sections 
Basic HTML front end design complete 
(no real function) 
'view my account' page HTML 
complete 
+ Add another card 
Done 
Solidify website idea (what it is, what it 
will be able to do) 
Research other websites for design 
deas 
Find the websites colour scheme 
First designs (XD, paper etc) 
Set of basic requirements from end 
user 
Talk to stakeholders bout designs 
Edit design and work to solidify a 
design 
+ Add another card 

Machine generated alternative text:
ChallengeM 
Things To Do 
Creating a new challenge page 
functioning (inserting a new challenge 
into the database) 
Creating sessions, i.e. Login page, and 
restricting the ability of certain users 
(intro of user level - admins, users etc.) 
When logged in, 'my account' page 
displays personal data (only seen by 
logged in user) 
+ Add another card 
6 
Doing 
Design databass 
Private 
OK 
Done 
First designs (XD, paper etc) 
Set of basic requirements from end 
user 
Talk to stakeholders bout designs 
Edit design and work to solidify a 
design 
Add users (admin, user) 
Basic HTML front end design 
complete (no real function) 
Leaderboard sections 
'view my account' page HTML 
complete 
another list 
Home page (with mini leaderboard 
and mini up and coming) complete 
and funcational 
Basic leaderboard and up and coming 
pages functioning (talking to database 
and displaying results in order) 
Upone selecting a user, displays user 
infomation (username, first name, 
wins, etc.) 
+ Add another card Machine generated alternative text:
ChallengeM 
Things To Do 
ebsiteæct e 
Doing 
Private 
OK 
Done 
complete (no real function) 
Leaderboard sections 
'view my account' page HTML 
complete 
Home page (with mini leaderboard 
and mini up and coming) complete 
and funcational 
Basic leaderboard and up and 
coming pages functioning (talking to 
database and displaying results in 
order) 
Upone selecting a user, displays user 
infomation (username, first name, 
wins, etc.) 
+ Add another card 
Nice to have 
-F Add a card 
Create admin section with links to 
restricted areas 
Ability for users to create a challenge, 
then accept or decline a pending 
challenge (adding challenge stages in 
e.g. complete, active, pending) 
When logged in, leaderboards 
display your name/data/stats in red 
Admin can view all pending events 
and accept/decline them 
When a challenge is created, invite 
(which can be accepted or declined) 
is only accessible to the 'none' 
creator involved 
+ Add another card 
Design databass 
Creating a new challenge page 
functioning (inserting a new challenge 
into the database) 
Creating sessions, i.e. Login page, and 
restricting the ability of certain users 
(intro of user level - admins, users etc.) 
When logged in, 'my account' page 
displays personal data (only seen by 
logged in user) 
+ Add another card 

Machine generated alternative text:
Cha engeM 
Things To Do 
Users, or admins can select a winner, 
thus taking it off the 'up and coming' 
Points, stats, wins etc. are now 
reflected by real results (e.g. a win will 
increases that users win count etc.) 
Login page / sessions all complete 
Ability for logged in users to edit their 
account details (username/names/bio 
etc.) 
Loggin, edit account, and register 
page checking for 'taken' usernames, 
and only allowing for usernames 
which aren't already exsiting 
+ Add another card 
6 
Doing 
Private 
• OK 
Create admin section with links to 
restricted areas 
Ability for users to create a challenge, 
then accept or decline a pending 
challenge (adding challenge stages in 
e.g. complete, active, pending) 
When logged in, leaderboards 2 
display your name/data/stats in red 
Admin can view all pending events 
and accept/decline them 
When a challenge is created, invite 
(which can be accepted or declined) 
is only accessible to the 'none' 
creator involved 
+ Add another card 
Done 
UdLdUdSe dilu ulsp1dY111Y results 111 
order) 
Upone selecting a user, displays user 
infomation (username, first name, 
wins, etc.) 
Creating a new challenge page 
functioning (inserting a new 
challenge into the database) 
Creating sessions, i.e. Login page, 
and restricting the ability of certain 
users (intro of user level - admins, 
users etc.) 
When logged in, 'my account' page 
displays personal data (only seen by 
logged in user) 
+ Add another card Machine generated alternative text:
ChallengeM 
Things To Do 
-F Add a card 
ebsite pr 
6 
Private 
OK 
Doing 
Design databass 
Users, or admins can select a winner, 
thus taking it off the 'up and coming' 
Points, stats, wins etc. are now 
reflected by real results (e.g. a win will 
increases that users win count etc.) 
Ability for logged in users to edit their 
account details (username/names/bio 
etc.) 
Loggin, edit account, and register 
page checking for 'taken' usernames, 
and only allowing for usernames 
which aren't already exsiting 
+ Add another card 
Done 
resulcteu erects 
Ability for users to create a challenge, 
then accept or decline a pending 
challenge (adding challenge stages in 
e.g. complete, active, pending) 
When logged in, leaderboards 
display your name/data/stats in red 
Admin can view all pending events 
and accept/decline them 
When a challenge is created, invite 
(which can be accepted or declined) 
is only accessible to the 'none' 
creator involved 
Login page / sessions all complete 
+ Add another card Machine generated alternative text:
ChallengeM 
Things To Do 
Admins can create challenges 
6 
ebsite pr*ct 
Doing 
Private 
OK 
Butler 
Design databass 
between two other users (admins can 
never be involved) 
Only logged in users can view pages 
(e.g. navbar directs to login page if 
session isn't started) 
Ability to add another admin account 
OR promote an existing account to 
an admin 
Admin ability to view all completed 
events and their outcomes 
Betting system designed (database) 
Betting HTML basic 
When a user/user enter a winner of 
the challenge, an admin must then 
Users, or admins can select a winner, 
thus taking it off the 'up and coming' 
Points, stats, wins etc. are now 
reflected by real results (e.g. a win will 
increases that users win count etc.) 
Ability for logged in users to edit their 
account details (username/names/bio 
etc.) 
Loggin, edit account, and register 
page checking for 'taken' usernames, 
and only allowing for usernames 
which aren't already exsiting 
+ Add another card 
Done 
restricted areas 
Ability for users to create a challenge, 
then accept or decline a pending 
challenge (adding challenge stages in 
e.g. complete, active, pending) 
When logged in, leaderboards 
display your name/data/stats in red 
Admin can view all pending events 
and accept/decline them 
When a challenge is created, invite 
(which can be accepted or declined) 
is only accessible to the 'none' 
creator involved 
Login page / sessions all complete 
+ Add another card 
Nice to have 
Ability for users to change their 
password 
Search bar for basic things (e.g 
leaderboard, users, events etc. 
+ Add another card 

Machine generated alternative text:
roards 
ChallengeM 
Things To Do 
ebsite pr 
ct e 
Doing 
6 
Private 
• OK 
Butler 
Admins ability to delete an account 
Admins ability to delete an event 
Logged in user able to delete own 
account 
Fix up the CSS and make all pages 
look good and related 
Betting system working and with 
relation to events and outcomes 
+ Add another card 
Design databass 
Betting system designed (database) 
Ability to add another admin account 
OR promote an existing account to an 
admin 
When a user/user enter a winner of 
the challenge, an admin must then 
approve of the result 
Ability for users to change their 
password 
+ Add another card 
Done 
Wiser etc.) 
Loggin, edit account, and register 
page checking for 'taken' usernames, 
and only allowing for usernames 
which aren't already exsiting 
Admins can create challenges 
between two other users (admins can 
never be involved) 
Only logged in users can view pages 
(e.g. navbar directs to login page if 
session isn't started) 
Admin ability to view all completed 
events and their outcomes 
Betting HTML basic 
+ Add another card 
Nice to have 
Search bar for basic things (e.g 
leaderboard, users, events etc. 
Dates displayed in words rather than 
numbers 
+ Add another card 

Machine generated alternative text:
Boards 
ChallengeM 
Things To Do 
ebsite pr 
å 
Private 
OK 
Butler 
+ Add anc 
Admins ability to delete an event 
Fix up the CSS and make all pages 
look good and related 
Make CSS for 'my account' section 
look good (done) 
Done 
secuull, d range u111ere[1L 
options 
Ability for users to change their 
password 
Logged in user able to delete own 
account 
ability for admin to delete other 
users account 
Ability to add another admin account 
OR promote an existing account to 
an admin 
Betting system designed (database) 
Admins ability to delete an account 
+ Add another card 
Nice to have 
Search bar for basic things (e.g 
leaderboard, users, events etc. 
Dates displayed in words rather than 
numbers 
+ Add another card 
OK 
Error catching on the login page (e.g. 
please make an account before etc.) 
nter a title for this card.. 
Doing 
Comments in code done 
Design database 
Betting system working and with 
relation to events and outcomes 
Betting/voting system working 
Resizing working on all key pages 
(admin and users) 
+ Add another card 
Add Card 
x 

Machine generated alternative text:
ChallengeM 
Things To Do 
Admins ability to delete an event 2 
Navbar changes between users and 
admins 
+ Add another card 
å 
Doing 
-F Add a card 
Private 
OK 
Done 
Comments in code done 
Design database 
Betting system working and with 
relation to events and outcomes 
Betting/voting system working 
Resizing working on all key pages 
(admin and users) 
Error catching on the login page (e.g. 
please make an account before etc.) 
Make CSS for 'my account' section 
Butler 
Nice to have 
Search bar for basic things (e.g 
leaderboard, users, events etc. 
Dates displayed in words rather than 
numbers 
+ Add another card 
look good (done) 
+ Add another card 
OK 

## Stage 3 – Ongoing testing and development

In this stage you need to provide evidence that you have had other people test your website and provide feedback – remember to look at both design and functionality as appropriate. Any changes you make based on that feedback should be explained. Evidence can be comments or screenshots.

To Achieve, you must complete at least **one** of these, whereas to provide evidence of **ongoing** testing you need to complete four or more. Feel free to do more!

Date: 26.05.2020

Student name: Jordan Bourke

Feedback:

* When you create and event or delete an event (as an admin) there should be some form of message that appears confirming that your request as received and acted upon. Currently it just redirects the user back to the home page with no indication of success or failure.
* When you’re logged out and click on options on the navbar, it just reloads the Login button with no message telling the user they must first Login

**Changes made as a result of feedback:** I created a section for both admins and users on the home page were messages will appear displayed text that tells the user if their request has been put through. This allows them to create an event, and then read the message telling them it’s been successfully created, without having to send the user to a new whole page.

I created some error messages on the Login page. This means if the user attempts to enter a page without first logging in, it displays an error message telling the user they must first log in.

Date: 26.05.2020

Student name: Alan Fu

Feedback:

* Admin panel sections all look horrible when you change the size of the page. This creates a bad user experience and takes away the ability to carry out some tasks on the site.
* Admins are able to create events between two of the same users (e.g. someone can challenge themselves)
* Is the Search box really required? What does it add that isn’t already easy to access?
* Description of how the site works/what it’s for is lacking on the homepage

**Changes made as a result of feedback:** I coded a lot more bootstrap in order to make my design responsive. I made it so a number of columns that content took up would change as the page shrunk so that at no point formatting would appear broken.

My website is now responsive for every page and so should work on almost all screen sizes. I have edited my code so that when admins try and create an event between two of the same users, an error message appears, and the event isn’t created.

I’ve taken the search box out as it’s not really required and due to a lack of time (as the site has so much other function I was focusing on) I decided to remove it.

On the home page I’ve added a ‘welcome’ text box. This switches to the admin panel if logged in as an admin. The successful messages occur here, as well as a brief overview of the site and it’s purpose etc.

Date: 31.03.2020

Student name: James Ecroyd

Feedback:

* You should use some more layering to give depth (lighter coloured panels are more elevated) and also consider using shadows to help with this too. Also, consider changing the navbar & footer colours as they’re very extreme.

**Changes made as a result of feedback:** Changed the ‘select user’ page, which he was referring to, so that now there are a range of shades, resulting in more depth and a better design. I’ve slightly lightened the navbar and footer, as it blends better with the rest of the site. This produces a better flowing site that appears more constant and easier to use/read.

Date: 08.06.2020

Student name: James Ecroyd

Feedback:

* When voting on an event, users are able to vote more than once on the same event.
* Admins shouldn’t be able to view their stats, considering they can’t take part in any challenges
* Inconsistency’s in red colouring for backgrounds
* Only the selectUser.php page, if you enter an invalid userID an error appears.

**Changes made as a result of the feedback:** I went through my sql and edited the code so that only users who weren’t already in the ‘bets’ table, with the same challengeID as the current challenge, could place a vote. This means users are only able to vote once on any given challenge. I edited my session code so that admins are unable to access the ‘events’ and ‘profile’ page when viewing their account information. This means the users won’t see any stats regarding their user wins and points. All background reds are the same colour. I introduced error catching so that if the userID is invalid, it redirects the user to the leaderborad.php page.

Date: 07.06.2020

Student name: Omri Kepes

Feedback:

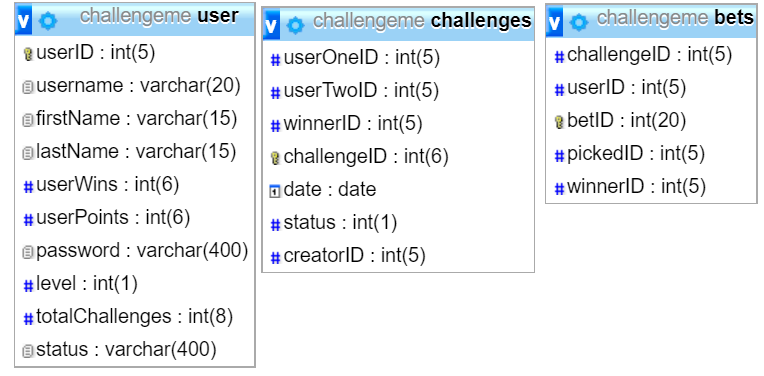
* When an event is created only the user who didn’t create the event, should have the option to accept or deny the challenge.
* Users shouldn’t be able to create an account with an already existing username and shouldn’t be able to change their name to a pre-existing username.
* Every status change when a user updates his/her own status. This is a really big floor.

**Changes made as a result of the feedback:** I went through my sql and php and decided in order to achieve this I must create a new column in my database. This was when I first introduced the ‘creatorID’ which allows me to only display the option to accept or decline the challenge, to the user involved that isn’t the creator (doesn’t have the creatorID). I wrote a new select query which checks the user table for any usernames that are equal to the ‘updated’ username, or ‘newly created’ username. If this query returned any results (and thus the username is already taken) an error message would show, and the username wouldn’t be updated/created.

I checked my php and sql and added a command so that when updating the users status, it only updates the status of the user whose userID is equal to the userID of the logged in user (e.g. Sessioned user). This meant the changes only applied to the correct user.

## Stage 4 – Final evaluation

Final database structure: Paste a screen shot of your database structure here from phpMyAdmin



Relevant implications: Explain how your design has addressed the relevant implications you identified above. Evaluate your final website against your relevant implications for Excellence.

**Privacy (Database):**

I website stores all passwords entered by the users (either current one or updated one) as a hashed/encrypted message (rather than plain text). The php hash key I’ve used is especially secure as it produces a different result every time, meaning it’s much harder to crack, thus more secure. This means the user information is kept well and safe, and in the unlucky case of a database breach, it’s difficult for the hacker to get the users individual passwords. This has meant I’ve successfully met the implication of privacy for this reason.

To add on this, I also assured that the text input areas where the user inputs their username and password is sql protected. This is done by using the real\_escape string, which removes any possible harmful inputs which could result in breaching of my database or site.

Collectively this approach to privacy allows for user data stored inside the database to be safe, confidential and secure.

As there are two key layers of protection against possible harmful actions, I believe I’ve successfully addressed the relevant implication of privacy and kept the user’s information and sensitive website data safe and secure.

**Future proofing (Database):**

My database successfully meets the relevant implication of future proofing. This has been achieved by assigning each user a userID to help avoid any possible name duplication errors and having these userID's automatically increase. This allows users to be added into the database quickly and simply and easily recognized within the code. Haven’t an Int max of length 5 means there are 100,000 different users able to be added, meaning if the site were to go life, and the number of users drastically increase, my database would still be able to handle this. Designing the database in such a way that it is the most efficient arrangement to increase the speed and amount of data being stored as also important and archived successfully. Assign all inputs appropriate max lengths was important as it allowed the site to be scalable and useable in the future.

I’ve also made sure to comment throughout my code, indicating the reason and function of specific sections. This allows for future editors to easily change or adapt the site which is important if it goes life in the future. I’ve used an external CSS page, allowing for easier aesthetic changes throughout the site, which helps for efficiency in the future if an editor wants to adapt the site.

I’ve used the most up-to-date CSS and HTML code, as well as Bootstrap resizing commands which makes it easier in future if someone wants to make changes to the website. I’ve made my site fully responsive which is important as users may which to view the site on either a big or small screen. In future this becomes more important as trends change, and users may begin to spend more time on the site on their phone.

For these reasons I’ve successfully met the implication of future proofing, and my site reflects this.

**Functionality (Website):**

My website has successfully met the relevant implication of functionality. This is because users are able to experience the required function to understand and take part in the websites purpose. Users are able to log onto their account, or create a new account, and enter the website.

There is significant error catching around the site, from ‘taken usernames’, to the ability to change one’s password, the site allows the user to easily and effectively carry out the tasks.

Users are able to be challenged, or create and challenge, while also bet on other challenges. There are different levels for admins and users, with admins being able to delete users, accept winners, challenges, and create more admins.

All links around the site are active and working, and the navbar and footer are accurate and work as they’re required. There is error catching in the admin section, with messages appearing should a challenge be created between two of the same users.

The site has been populated with lots of error catching on every page to deal with unexpected inputs. When a get array is not set, or page is linked that doesn’t exist, the site will display a 404 error as error catching to tell the user that the page, they are looking for does not exist.

All these error catching functions further improves to the site’s overall functionality.

The website has a responsive design that changes for different screen sizes so that everything on the website is still visible and works. This also meets the implication as it makes the website function well on small screens.

My site doesn’t offer the function or ability to comment on a forum (which was mentioned in the end user requirements), due to a lack of time when producing the site. If given more time, this feature could easily be introduced, as the basic database formation supports it, and beta html code has already been produced. This isn’t a big deal considering all the features and function of the site, so adding more isn’t as important.

Overall, my site meetings this implication as it carries out all the required tasks effectively and easily, while error catching and acting on unexpected results given. It also resizes and appears easy to use.

**Usability (Website):**

My website does a very good job at meeting the implication of usability by making sure that it meets many of the usability heuristics.

There is a consistent feel across the website on all pages, which makes it easier for the user to navigate around the site. Basic links are all located in expect areas e.g. the navbar, admin panel etc. making to site more intuitive and easier to user. The same style is used for call buttons, backgrounds, fonts etc. This helps to better the user experience and site usability.

By assuring important pages like the logout, home and leader board pages are clear and easy to locate helps improve the sites usability. All the buttons are clearly labelled and obvious to their function.

My site is coded and designed in such away that it’s responsive and useable from a range of different screen sizes. This means on phones, tablets, laptops etc. the website still carries the same ‘feel’ and function. This helps improve the usability of the site in general, as the user isn’t held back by the size of their screen. Users are able to get feedback on their actions (e.g. successful creation of a challenge) which helps eliminate any confusion the user may have while operating the site. This further improves the sites usability.

**Overall to evaluate:**

My site thoroughly addresses all chosen implications extremely well. For privacy it keeps the user’s information secure and hashed, while for functionality my site provides a large list of tasks users can carry out, along with a responsive design, and error catching to improve the sites usability. Future proofing was addressed when determining the database max limits on things like userID’s, along with commenting in code, and primary autoincrement keys. Finally, my site fully addresses the implication of usability through many areas, for example error messages, which all result in my site having very good ease of access.

End-user requirements: Explain how your design has addressed the purpose and end-user requirements you identified above. Justify decisions for Excellence.

Jordan:

Date: 8/6/2020

* Site is easy to use, and users are able to carry out tasks required.

Alan:

Date:8/6/2020

* Good flow around the whole site, easy to use and create and accept challenges.

James:

Date:8/6/2020

* Easy to use and carry out challenge. Admin section is simple and easy to use.

I’ve successfully met the end User requirements because after end user testing, and students who have used the website were able to carry out the required function which was the purpose for the site.

When user testing, James, Alan and Jordan all were easily able to create a challenge, challenge other users, be challenge, and bet on other events. This was done with easy navigation around the site, and with the help of the error messages, with the correct amount of feedback on the site.

Because these end users were able to meet the required function for the site, and this could be carried out relatively frictionless, its fair to conclude that my site successfully meets the relevant implications of usability and functionality.

The users commented that resizing on the site was good, and the user experience was positive.

Each user was able to log into their account, or create a new account if required, and their personal passwords were all kept secure and hashed, which is a lot safer. This, along with all users being assigned a new UserID shows that the site meets this implication.

User testing throughout the site development allowed me to make changes and additions to the site, which resulted in a more effective and usable site.

An example of this is the addition of the creatorID which was added, along with design changes, functional errors etc. The feedback which was received throughout the process helped me better meet these requirements.

A part of my website I changed because of user testing was improving and adding various feedback messages across my website to help explain to the user what is happening, to improve ease of use by reducing any confusion for the user. This change helped improve my usage of both the ‘Help users recognise, diagnose, and recover from errors’ and ‘Visibility of System status’ heuristics.

**Applications that helped the development of the site.**

I used two key 3rd party applications when making this site. The first was trello which allowed me to easily work on specific tasks on the site, and the other being Adobe XD which allows me to easily visualize my design early on, as well as get feedback on it quickly.

Trello allows me to separate tasks required in the development of the site. This was done by creating a list of ‘to do’ tasks, and sections such as ‘doing’ and ‘done’. This meant I could work on multiple tasks easily and clearly, without getting confused or mixing code. Trello also made sure my site was keeping on track with dates and tasks, and assured id always know what to work on next. Thus, came handy when I ran into problems during lockdown, as I couldn’t work on a particular section of the site, so I stopped, and started another section. This was made easy due to my use of trello. Screenshots above show my progress along with trello usage.

Trello also allowed me to separate the database tasks, design, and basic html code tasks, and order them. This made it much easier to manage and work on as I broke the whole site into smaller tasks.

Adobe XD allowed us to easily visualize our designs and share them which made it easier to get feedback and change. This, rather than using pen and paper to create designs, meant we could share it to others for feedback, and make changes easily and quickly. This allowed us to create a very solid design early on, which made it much easier when coding our site as we were able to refer to a template for what it should look like.

Without XD It would’ve been much harder to show our designs to possible users and get feedback on them, as cresting them would have otherwise been very time consuming and difficult.