**Importance of hashing a password**

Hashing a password is an important feature for the security of systems and applications. For SetStats the user will need to provide a username and email, so the password needs to be hashed to prevent unauthorized access to this. Hashing itself is merely the process of taking the password of a user and then scrambling it into a short string made up of letters and numbers as seen below. The hashing function is essentially an algorithm that scrambles the plaintext password into a random string of bits.

**md5(helloworld) = fc5e038d38a57032085441e7fe7010b0**

**Diagram

Description automatically generated**

**Process of hashing a password**

The process of hashing starts when a user goes onto SetStats and joins by signing up with their username and password. Once a password is sent through a hash function that hash then will be stored in our database. Then when a user has to login again they type in their password again and their password goes through the same hashing function as seen before. On the server side a check will commence to match the hash against the one already stored in the database for the user. Once the check is complete and it can be verified that both hashes match up then the user is granted access to SetStats. With the storing of passwords as hashes opposed to plaintext it as an important feature because it gives us the opportunity to no know of the password itself but just that the user has entered the correct one. The hash in the database is all that is needed to verify and check the password. In the event that a user forgets their password we can send them a email or message enabling them to create a new one. With hashing there is no need to have knowledge of the old password.

**Example diagram of process of hashing password**

Diagram

Description automatically generated

**Important features for strong password security**

Some of the most important features needed to create a strong password include:

* Creating a long random password that includes numbers, letters, and symbols. This makes it harder to crack the password.
* Regularly changing that password every 6 months to a year. Updating it often makes it harder to guess for hackers.
* Only use 1 password per website. Never reuse the same password on multiple websites.
* The minimum length of passwords should be at least 8 characters in length but ideally longer.
* Never use dictionary words as they are more prone to being cracked by a hacker.
* The password should not include any personal information like the place you are from, names of family members, your favourite sports team etc.
* Consider using a password manager to store all your passwords in a safe place.

**GDPR Requirements**

**Graphical user interface

Description automatically generated with medium confidence**

After GDPR came into effect in May 2018 it changed the rules and law around data and privacy rights. The need to ask users for consent when storing and processing their data is the norm now and must be adhered to. With every user now needing a name and password to sign up or join a website or application it is more important than ever to follow the guidelines and rules that come with GDPR.

One of the key statements from GDPR with personal data means that under GDPR all personal data must be processed “in a manner that ensures appropriate security of personal data including protection against unauthorized or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organizational measures.”

Although not specific the appropriate part of the text to me is subjective but it does include password security which hashing covers. The risk of hefty fine for noncompliance or even failure to address any security concerns says to me that having SetStats hash users’ passwords both complies with GDPR and leaves us not at risk of a potential fine or issue with the regulations. It is in our best interests as developers of our application and from an ethical perspective to all of our users to have too much security measures rather than having too little or none at all. In keeping with the GDPR compliance of stopping unauthorized individuals from getting access to data or resources our password policy should match this by having the same goal. This means a strong password policy to keep in line with the regulations and hashing every password for each and every user.

#### 