Security is a big issue obviously. Especially so when you are dealing with money and people’s careers. I have spent some time investigating how to implement sufficient security into the platform to make it both easy to retrieve data but not be too vulnerable to the many attacks that take place in the cyber world.

Good practice would suggest encrypting the password and then masking it before saving it on the databases if possible.

It seems that when storing a password there are many bad options that you can go for but some seemingly good ones as well.

1. Store the password as plain text.
2. Encrypt the password. Save the private and public keys somewhere, potentially away form the main sever and you might be fine
3. Use hashing. A simple has of the password and potentially the username is good
4. You can go further with the hashing buy using salting. This is when you a series of letters to the end of the hash, either random or predefined. Either for each user or a specific number used throughout the organization. Never use low quality salts like a simple counter or a random number. Can use something like Microsoft’s Crypto API.
   1. Sha-256? Porobably not if you can help it. Not safe against GPUs/
   2. Bcyrpt (used by Auth0)
   3. PBKDF2
   4. Hmac-sha-256

\*\* DO NOT USE YOUR EMAIL PASSWORD NUDGE

\*\* COMPARE THE NEW CREDENTIALS AGAINST THE DATABASES OF BREACHED CREDENTIALS

\*\* RATE LIMITING (THE NUMBER OF TIMES AN ATTACKER CAN TRY ACCESS THE SITE)

Another important point to make is around the transmission of data. It seems it necessary to encrypt the data, or perform some other security measure, beyond just a simple MD5 like checksum, to protect the data.

TLS with 128-bit AES is what Auth0 uses for the data encryption at rest and in transit. See the article [here](https://auth0.com/blog/hashing-passwords-one-way-road-to-security/).

Questions

1. Does the user get an email if a login takes place on a different device or a suspicious location?
2. Can users create a physical security key (USB) that is used to authenticate?
3. The use of other standard 2-factor authentication measures? (OTP, APP)
4. How would we use something like Microsoft Active Directory (AD) for authentication?
5. How is easy is it to change all the password even if we don’t know all the passwords?
6. What happens to employees’ passwords when they leave the company?
7. Do we need another layer of security that is performs authentication? (see the likes of Spotify and their authentication set up)