Agenda

1. What is authentication
2. What is authorisation
   - RBAC
3. How auth flow works.

What is authentication / Authorization.

scaler.com/academy
leetcode.com/problems
} No need of login

Authentication

scaler.com/meetings/i/_____
leetcode.com/problems/10/submit
} You need to confirm who you are.

Authorization

scaler.com/admin/_____
leetcode.com/problems/10/edit
} only admins can visit

Another ex:

Consider you're at an school gate

i) Anyone with valid 'id' can enter the school

ii) Only authorised people can enter principal room.

i) → Authentication.

ii) → Authorization.

Authentication

(i) A concept by which a website will identify a user.

(ii) Tell me who you are and you are allowed to visit.

Authorization

Tell me who you are
+
Do you've appropriate permission }

Book My Show

	Authentication	Authorization.
i) See availability of a particular show	X	X
ii) book a seat	✓	X
iii) cancel a ticket	✓	✓

General ways of authentication for websites.

(i) email + password

(ii) phone + otp.

RBAC : Role Based Access Control / Authorisation.

scaler.com/topics ⟶ anyone can visit

scaler.com/topics/course/10 ⟶ " "

Click on a video, ⟶ scaler.com/topics/course/10/video/1.

Tell me who you are ---

Link to admin page.

Tell me who you are ---

+

You should have necessary roles.

users

id	---	---

roles

rId	name

user-roles (M:M)

uId	rId

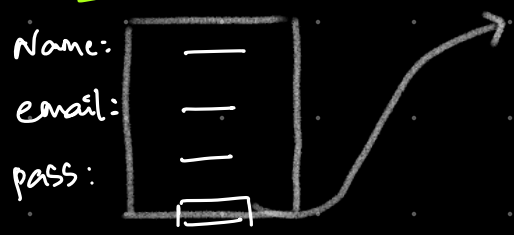
How auth flow works).

Websites want to uniqe identify people; email / phone.

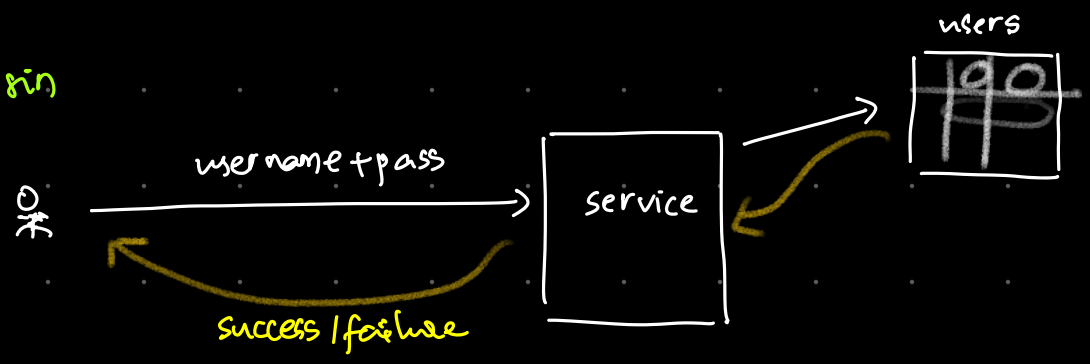
How to validate ?

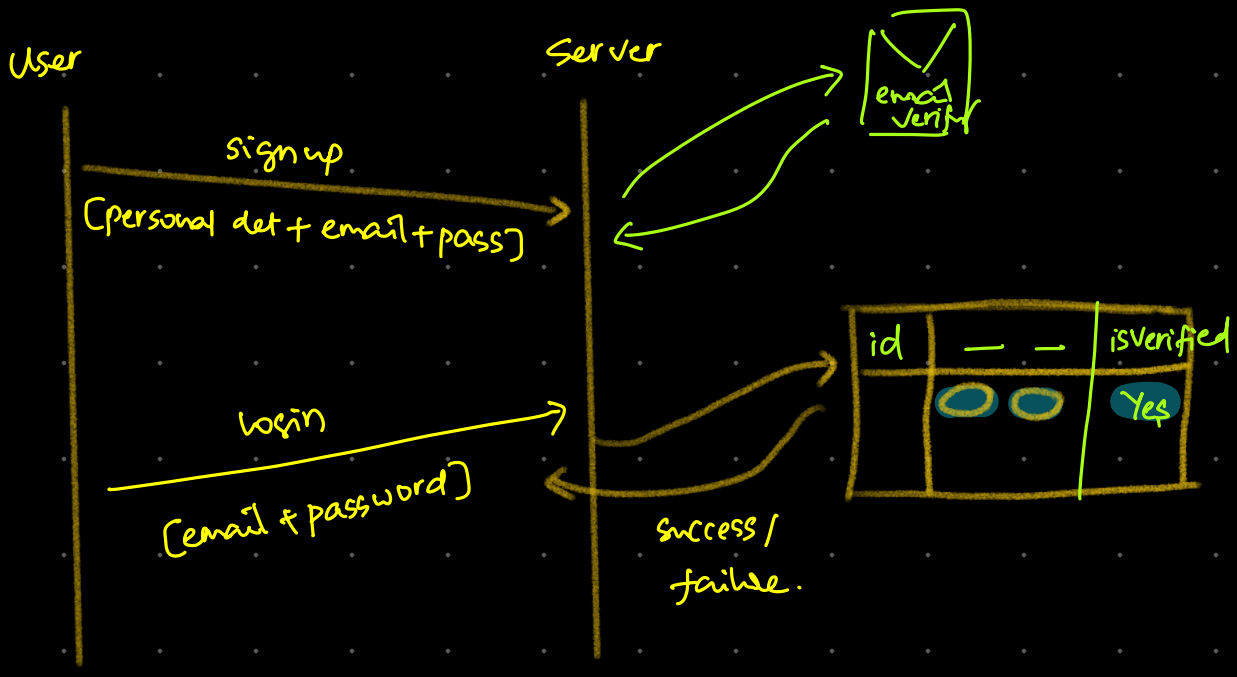
password / otp.

① Sign up



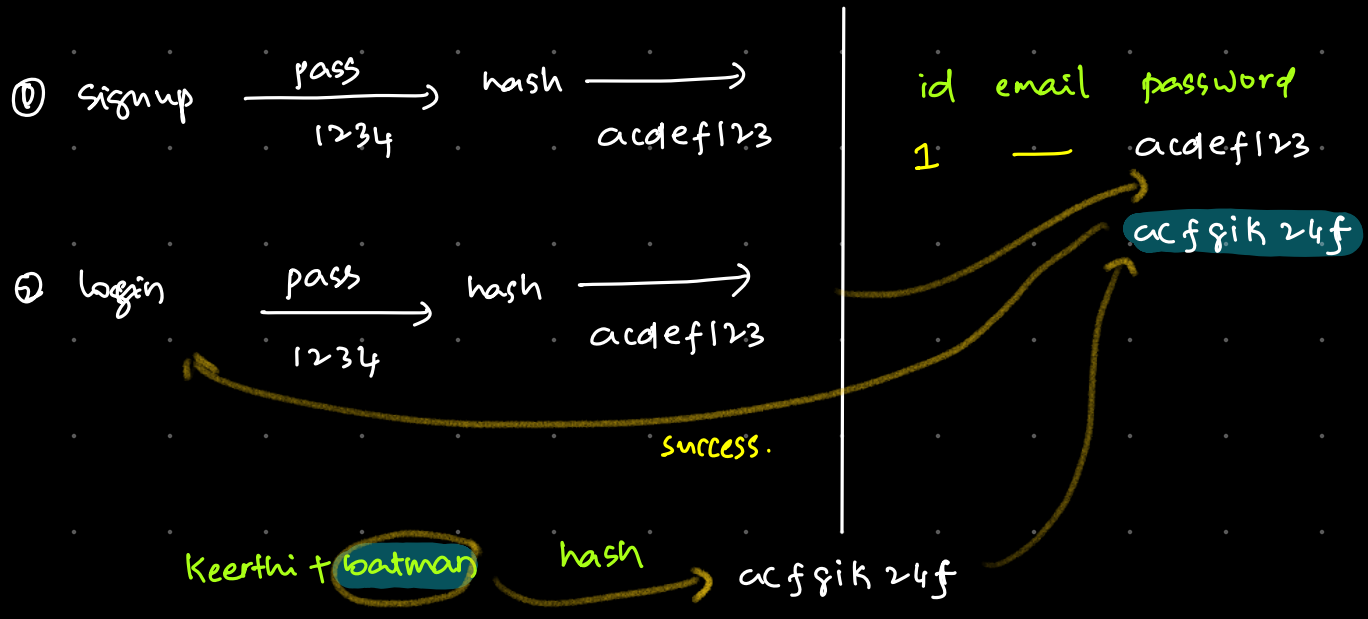
① login





is it ok to store password as string? NO--

Sol^n: Store password as a hash



If the same password results in same hash, then password can be leaked easily.

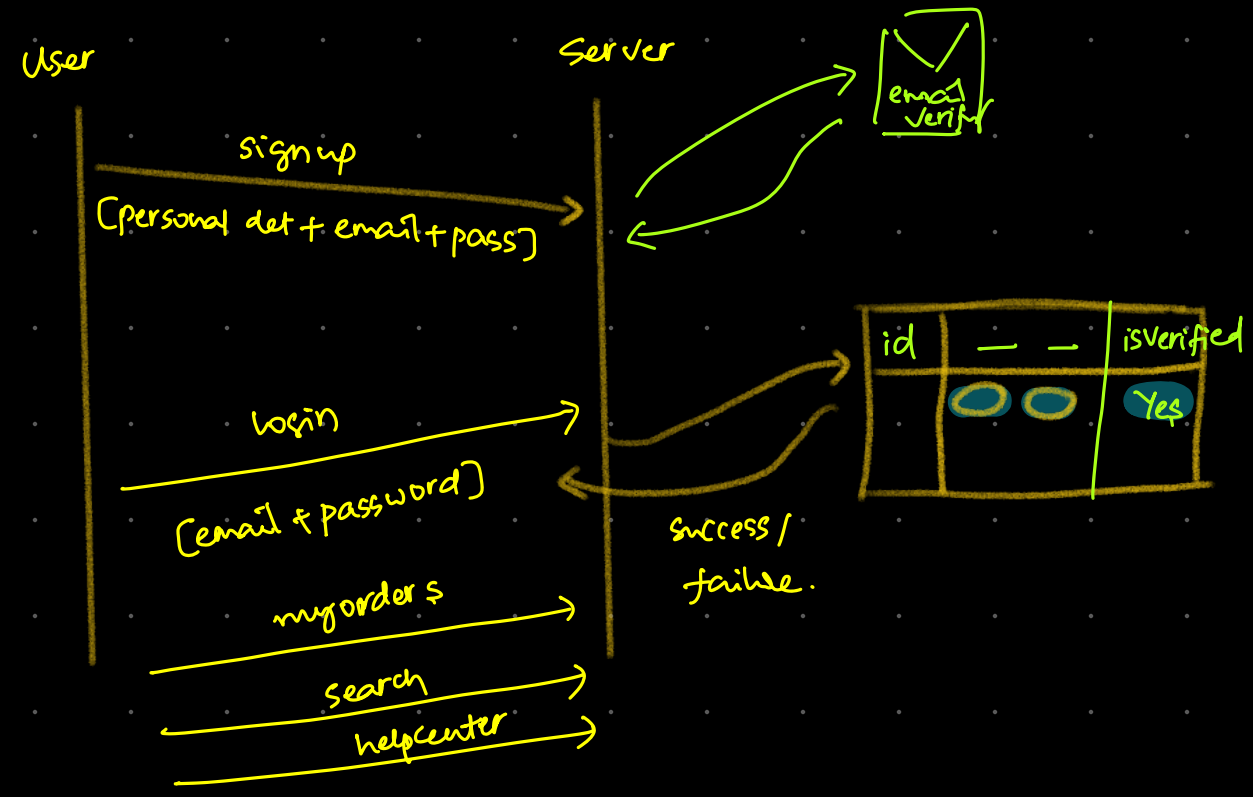
Sol^n: [password + salting]
↓
[email + time + served_id] etc

Bcrypt encoder

1. .encode()
2. .verify()

1. You cannot decode the password.
2. Same password will result in diff hash.

Any issues?



There's a huge load on login service

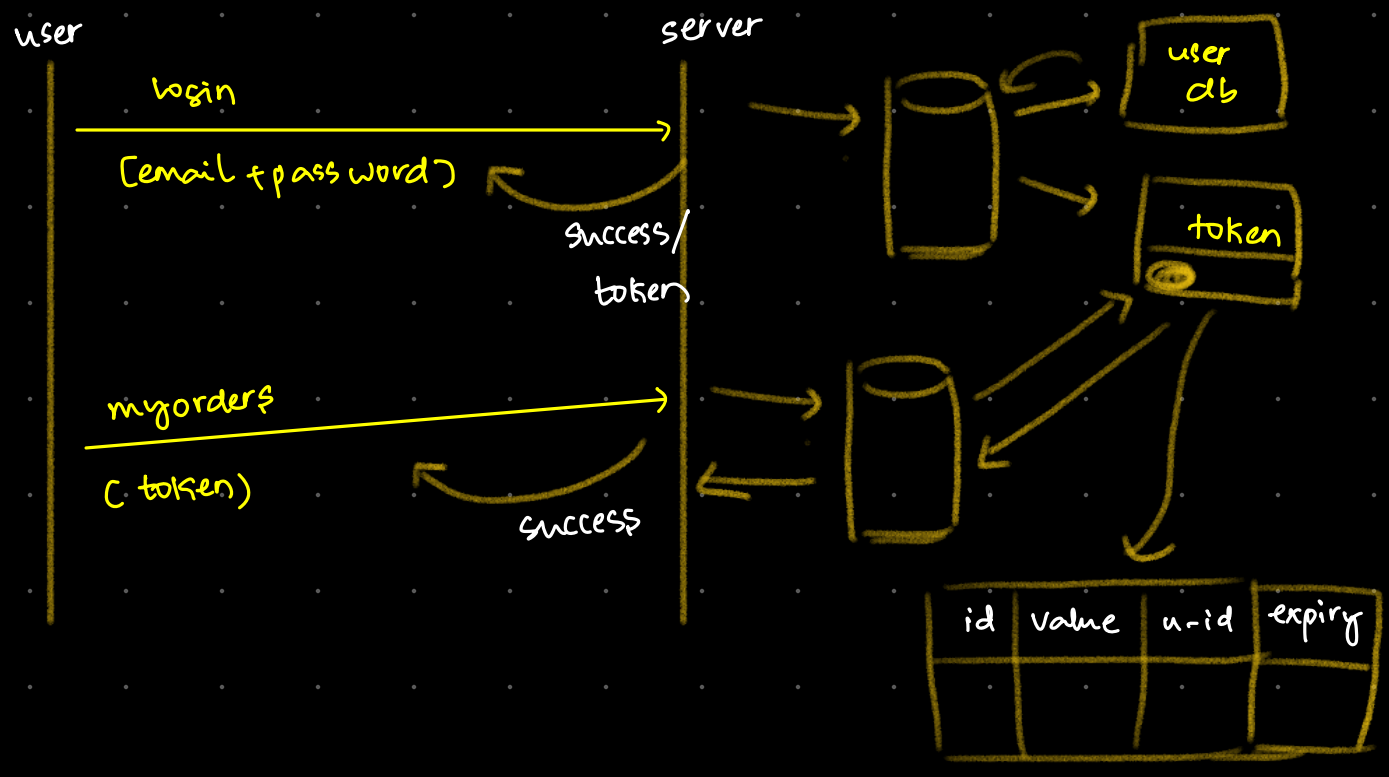
1. will make a db call
2. will run bcrypt.

This will slow down the throughput.

Sol: Inspired from 'badge' concept in restaurants / resorts.

We don't authenticate people everytime, instead we give them something, if they show us that they're authenticated & authorised.

Badge → Token



token → temporary password.

here, we're still hitting the db.