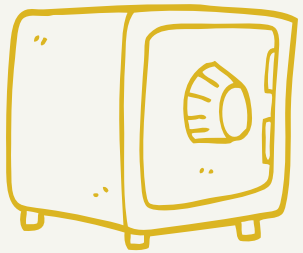
A U T H

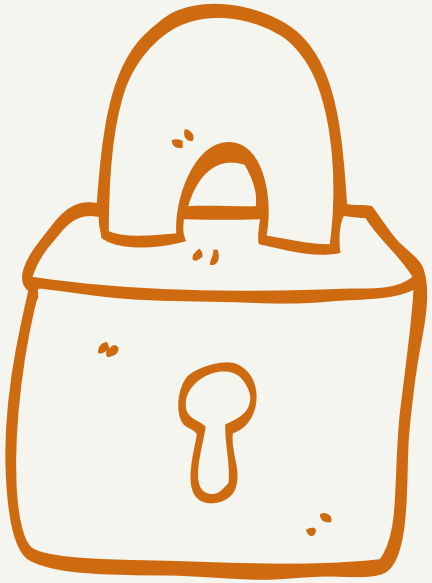


WEB DEVELOPER
BOOTCAMP

DEEP BREATH. IT'S TIME FOR AUTH.

DEEP BREATH. IT'S TIME FOR AUTH.

DEEP BREATH. IT'S TIME FOR AUTH.



AUTHENTICATION

WHAT IS IT?

Authentication is the process of **verifying who a particular user is.**

We typically authenticate with a username/password combo, but we can also use security questions, facial recognition, etc.

DEEP BREATH. IT'S TIME FOR AUTH.

DEEP BREATH. IT'S TIME FOR AUTH.

AUTHORIZATION

WHAT IS IT?

Authorization is verifying what a specific user has access to.

Generally, we authorize after a user has been authenticated. "Now that we know who you are, here is what you are allowed to do and NOT allowed to do"

DEEP BREATH. IT'S TIME FOR AUTH.

RULE #1

NEVER STORE PASSWORDS

DEEP BREATH. IT'S TIME FOR AUTH.

RULE #1

NEVER STORE PASSWORDS

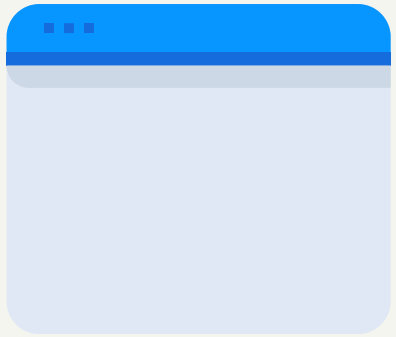
```
{
    username: 'kittycatluvr',
    password: 'meowmeow999!'
},
{
    username: 'geckoGuy',
    password: 'lizard987'
}
```

DEEP BREATH. IT'S TIME FOR AUTH.

DEEP BREATH. IT'S TIME FOR AUTH.

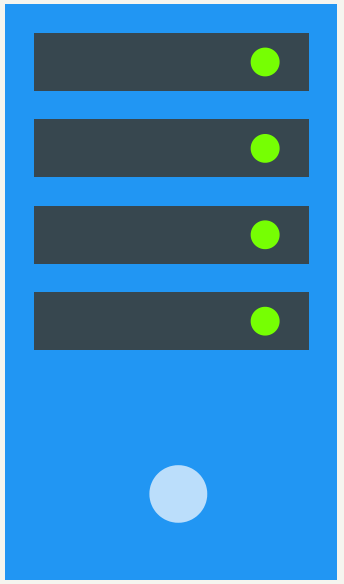
DEEP BREATH. IT'S TIME FOR AUTH.

CLIENT



LOG ME IN WITH:
Username: 'geckoGuy'
Password: 'lizard987'

SERVER



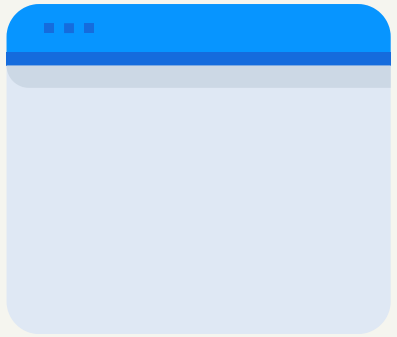
DATABASE

```
{
    username: 'kittycatluvr',
    password: 'meowmeow999!'
},
{
    username: 'geckoGuy',
    password: 'lizard987'
}
```

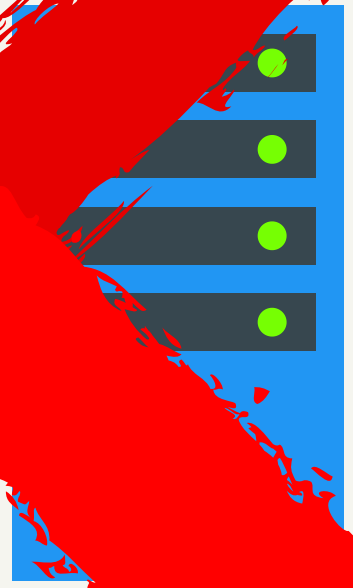
DEEP BREATH. IT'S TIME FOR AUTH.

DEEP BREATH. IT'S TIME FOR AUTH.

CLIENT



LOG ME IN W
Username: 'geck
Password: 'lizard



DATABASE

{

username: 'kittycatluvr',
password: 'meowmeow999!'

},
{

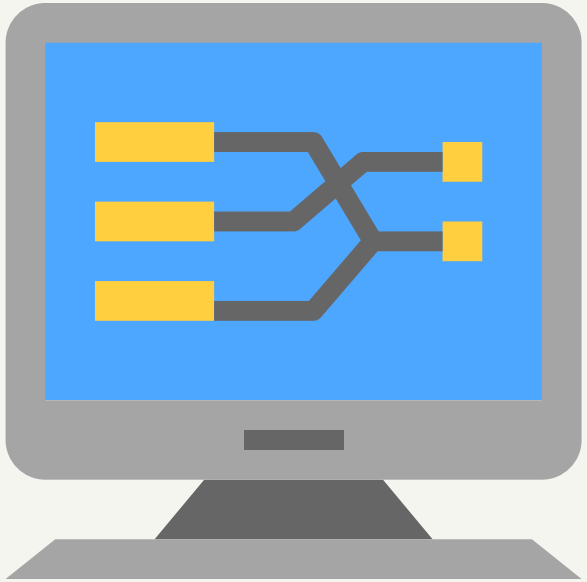
username: 'geckoGuy',
password: 'lizard987'

}

DEEP BREATH. IT'S TIME FOR AUTH.

NO!

DEEP BREATH. IT'S TIME FOR AUTH.



HASHING

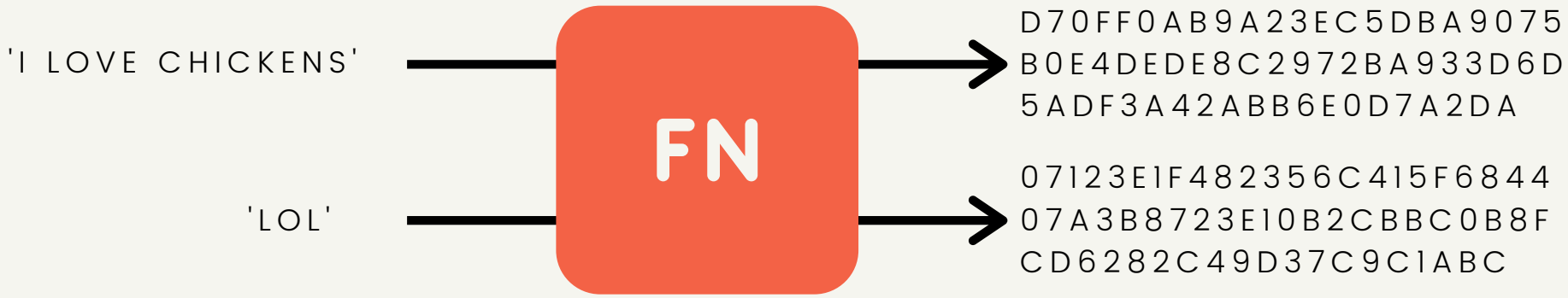
THE SOLUTION!

Rather than storing a password in the database, we run the password through a **hashing function** first and then store the result in the database.

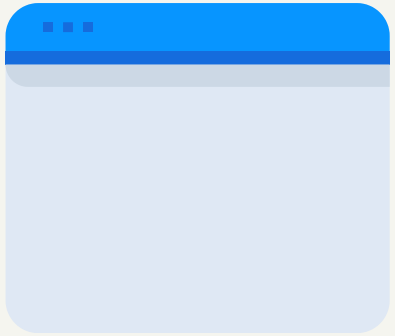
DEEP BREATH. IT'S TIME FOR AUTH.

HASHING FUNCTIONS

Hashing functions are functions that map input data of some arbitrary size to fixed-size output values.

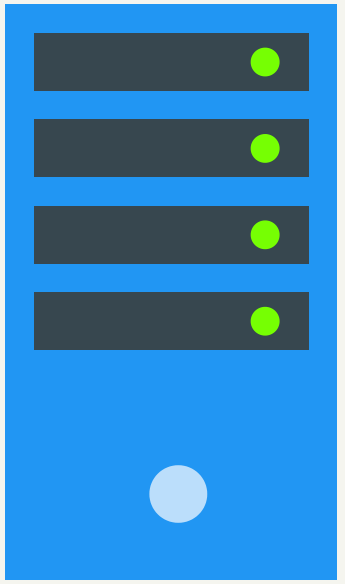


CLIENT



LOG ME IN WITH:
Username: 'geckoGuy'
Password: 'lizard987'

SERVER



DATABASE

{

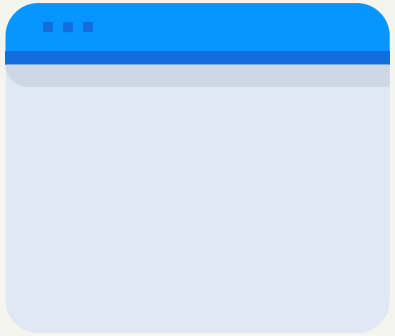
username: 'kittycatluvr',
password:'d70ffoab9a23ec5dba9075b0e4de
de8c2972ba933d6d5adf3a42abb6e0d7a2da'

},
{

username: 'geckoGuy',
password:'07123e1f482356c415f684407a3b87
23e10b2cbbc0b8fcd6282c49d37c9c1abc'

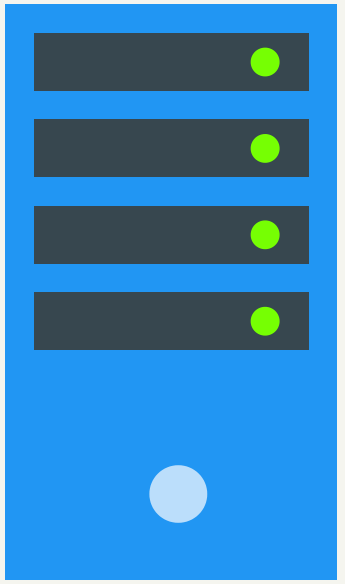
}

CLIENT



LOG ME IN WITH:
Username: 'geckoGuy'
Password: 'lizard987'

SERVER



DATABASE

{

username: 'kittycatluvr',
password:'d70ffoab9a23ec5dba9075b0e4de
de8c2972ba933d6d5adf3a42abb6e0d7a2da'

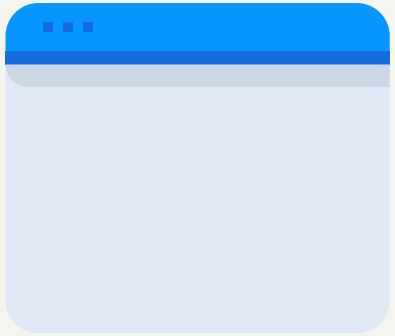
},
{

username: 'geckoGuy',
password:'07123e1f482356c415f684407a3b87
23e10b2cbbc0b8fcd6282c49d37c9c1abc'

}

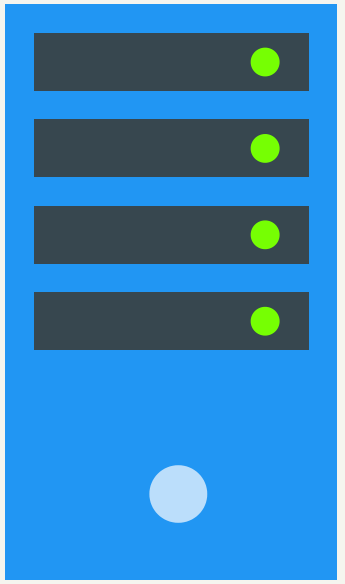
'LIZARD987' → FN →

CLIENT



LOG ME IN WITH:
Username: 'geckoGuy'
Password: 'lizard987'

SERVER



DATABASE

{

username: 'kittycatluvr',
password:'d70ffoab9a23ec5dba9075b0e4de
de8c2972ba933d6d5adf3a42abb6e0d7a2da'

},
{

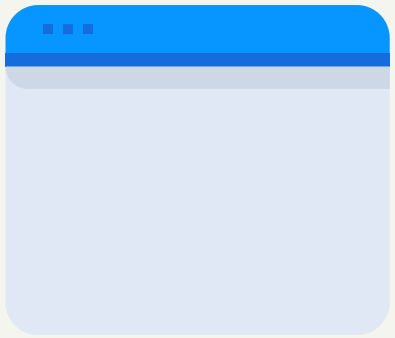
username: 'geckoGuy',
password:'07123e1f482356c415f684407a3b87
23e10b2cbbc0b8fcd6282c49d37c9c1abc'

}

'LIZARD987' → FN →

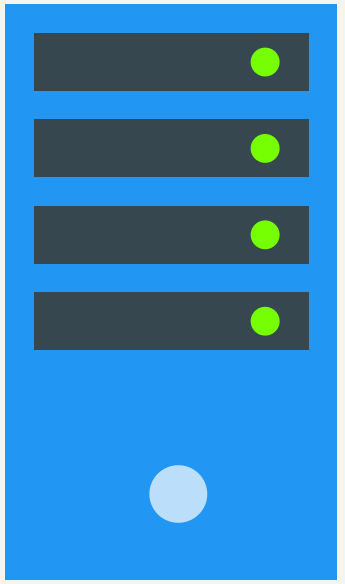
07123E1F482356C415F6844
07A3B8723E10B2CBBC0B8F
CD6282C49D37C9C1ABC

CLIENT



LOG ME IN WITH:
Username: 'geckoGuy'
Password: 'lizard987'

SERVER



DATABASE

{

username: 'kittycatluvr',
password:'d70ffoab9a23ec5dba9075b0e4de
de8c2972ba933d6d5adf3a42abb6e0d7a2da'

},
{

username: 'geckoGuy',
password:'07123e1f482356c415f684407a3b87
23e10b2cbbc0b8fcd6282c49d37c9c1abc'

}

'LIZARD987' → FN →

IT'S A MATCH!

07123E1F482356C415F6844
07A3B8723E10B2CBBC0B8F
CD6282C49D37C9C1ABC

DEEP BREATH. IT'S TIME FOR AUTH.

CRYPTOGRAPHIC HASH FUNCTIONS

1. One-way function which is infeasible to invert
2. Small change in input yields large change in the output
3. Deterministic - same input yields same output
4. Unlikely to find 2 outputs with same value
5. Password Hash Functions are deliberately SLOW

DEEP BREATH. IT'S TIME FOR AUTH.

DEEP BREATH. IT'S TIME FOR AUTH.

S A L T S

AN EXTRA SAFEGUARD

DEEP BREATH. IT'S TIME FOR AUTH.

DEEP BREATH. IT'S TIME FOR AUTH.

PASSWORD SALTS

OMG THAT'S SO RANDOM!

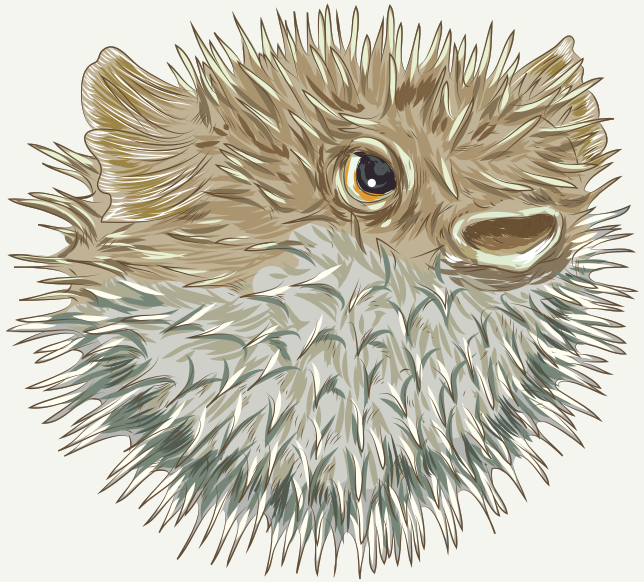
A salt is a random value added to the password before we hash it.

It helps ensure unique hashes and mitigate common attacks

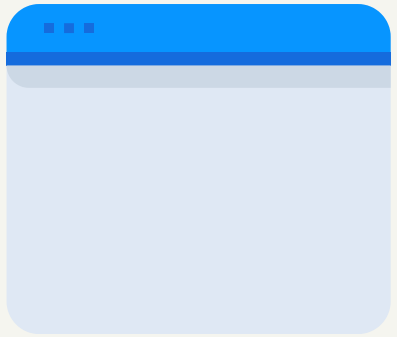
DEEP BREATH. IT'S TIME FOR AUTH.

B C R Y P T

OUR HASH FUNCTION!

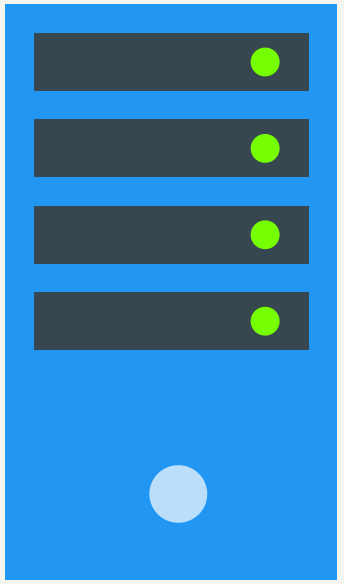


CLIENT



I have a cookie for you!
Session ID is 4

SERVER



DATA STORE

```
{
    id: 4,
    shoppingCart: [
        {item: 'carrot', qty:2},
        {item: 'celery', qty:5},
        {item: 'taser;', qty:99},
    ]
}
```

DATA STORE

```
{
    id: 3,
    shoppingCart: [
        {item: 'lime', qty:1},
        {item: 'la croix', qty:99},
        {item: 'lemon', qty:2},
    ]
},
{
    id: 4,
    shoppingCart: [
        {item: 'carrot', qty:2},
        {item: 'celery', qty:5},
        {item: 'taser;', qty:99},
    ]
},
{
    id: 5,
    shoppingCart: [
        {item: 'apple', qty:2},
        {item: 'onion', qty:5},
        {item: 'pear;', qty:9},
    ]
}
```