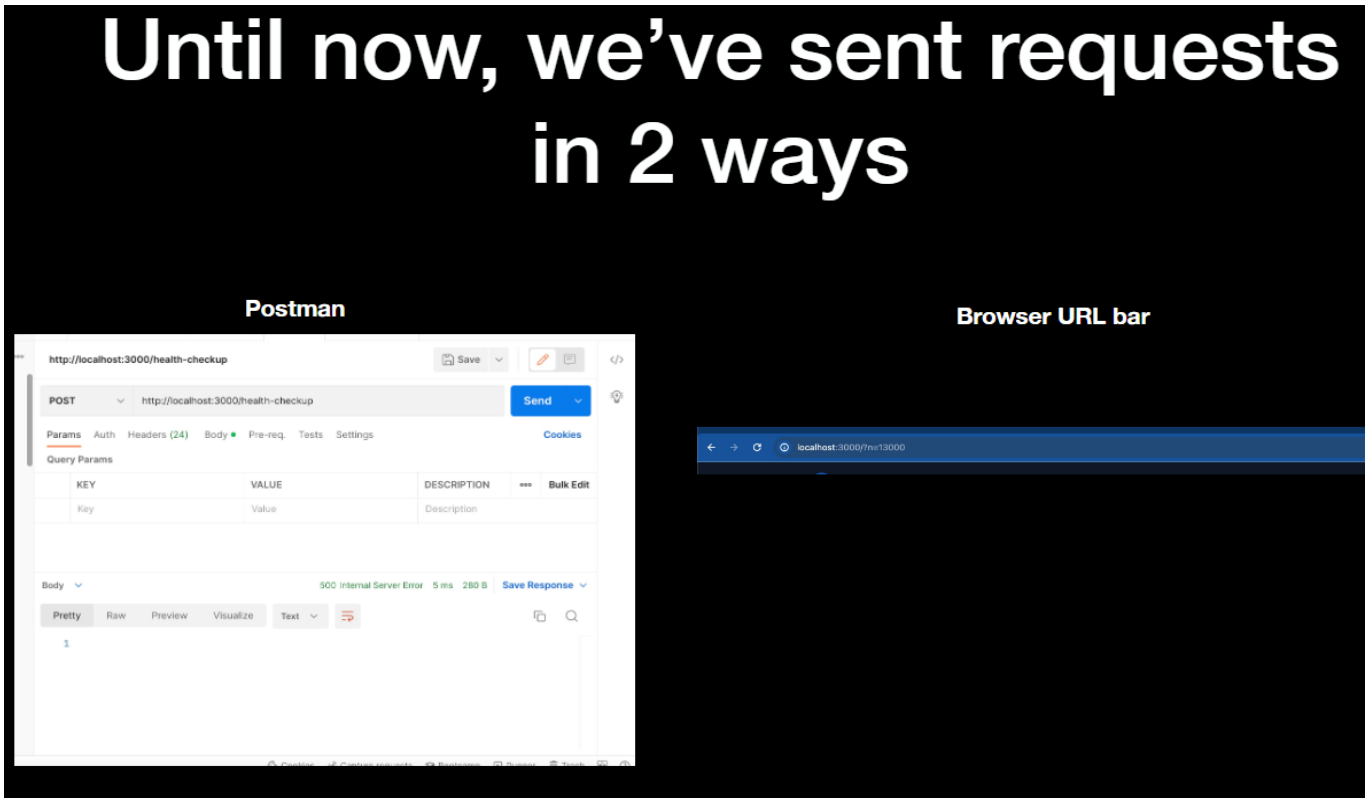
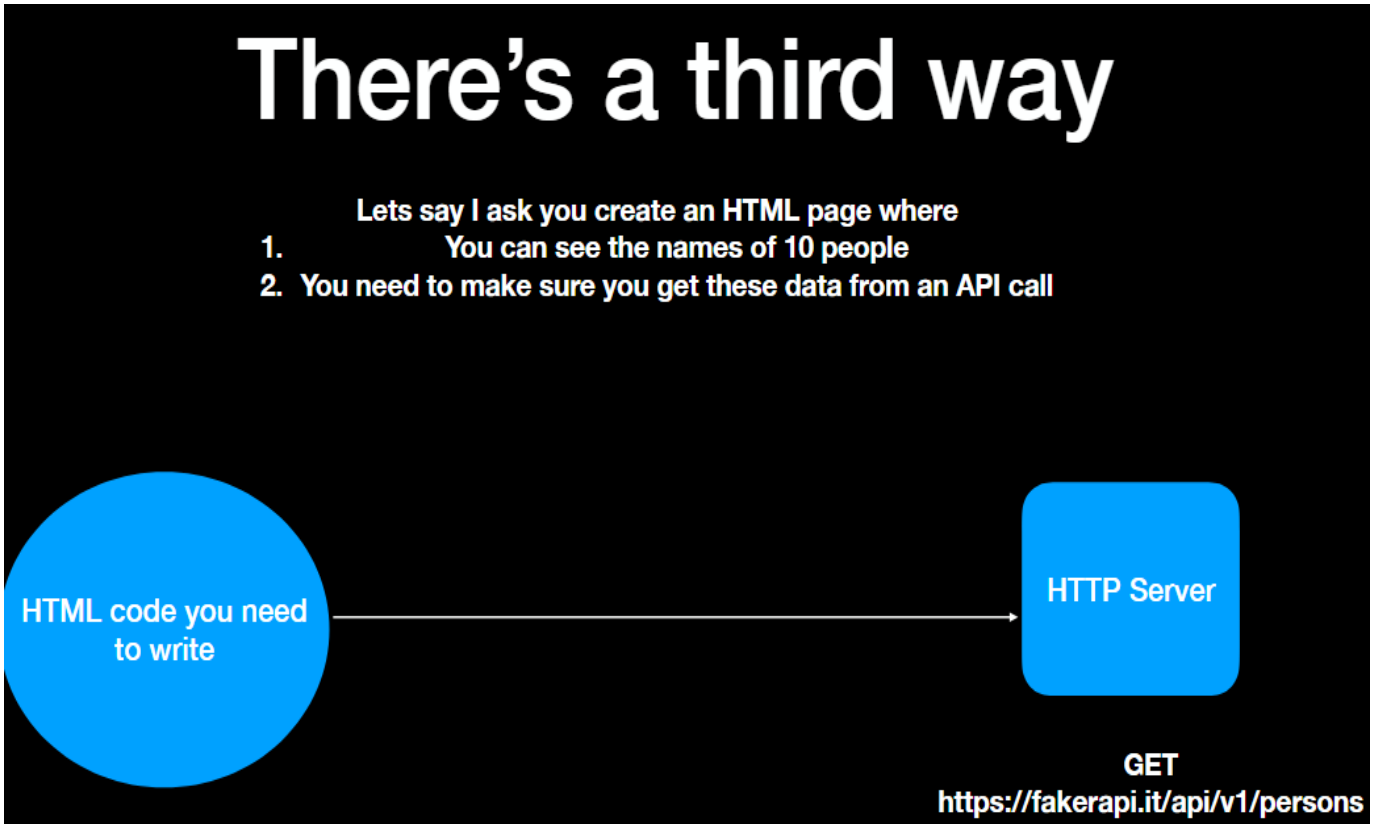
Dynamic Backend Authentication using Express, JWT and MongoDB

First, we will learn about fetch API.



The 1st one is a POST request and the 2nd one is a GET request.



In the earlier two ways we were posting or getting the data on the server side only, but now we have to take those data from the server into the html/frontend side.

Like we are using someone else backend to get the data so for that we use (fetch) method.

Let me show you :-

```html
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>replit</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>

<body>
  <button onclick="getAnimalData()">Get animal data</button>
</body>

</html>
```

This function is written in an index.js file.

Now, I will show how fetch works here:-

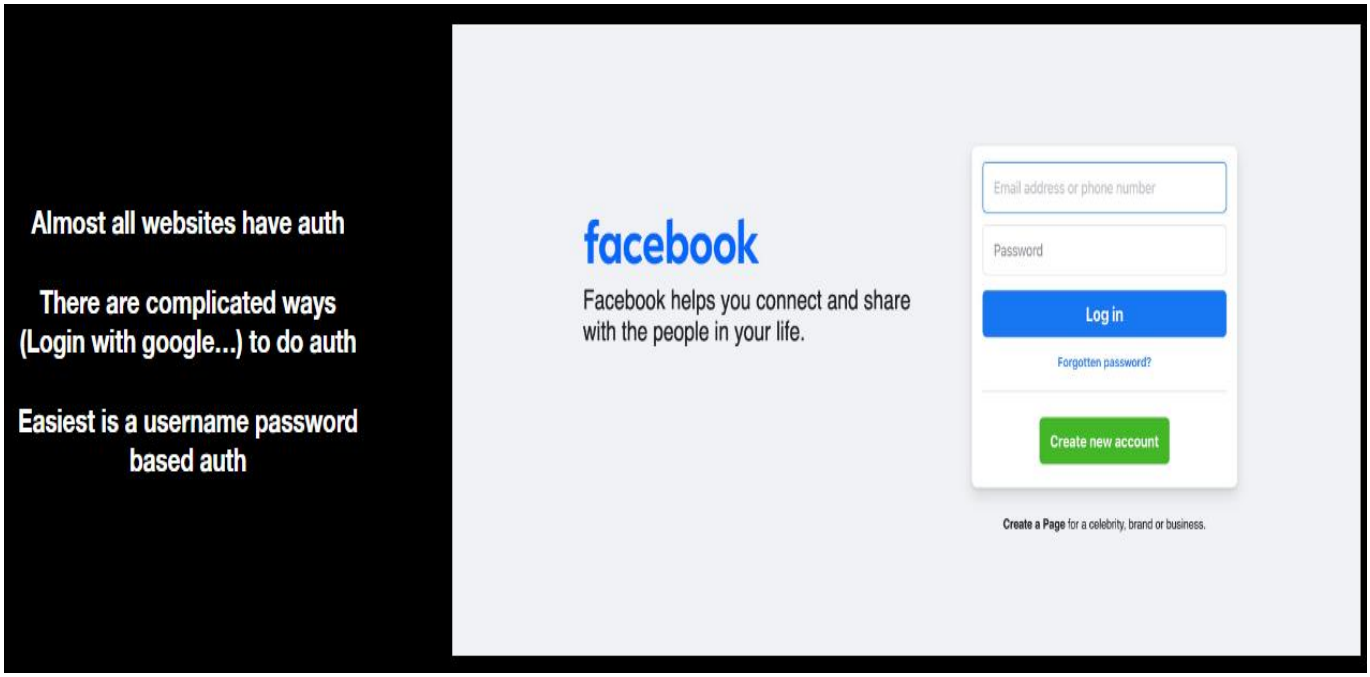
```html
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>replit</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>

<body>
  <div id="container">
  </div>
  <button onclick="getAnimals()">get animals</button>
  <script>
    function getAnimals() {
      fetch("https://fakerapi.it/api/v1/persons")
        .then(async function(response) {
          const jsonData = await response.json();
          document.getElementById("container").innerHTML = JSON.stringify(jsonData.data);
        })
    }

  </script>
</body>

</html>
```

Ye fetch ke andar jo karnama kia h , wo krne ki zaroorat nhi h , normal tarike se bhi ho jata h.

So basically we use fetch to take data from another person's server.

Now we will learn about Authentication in detail.

AUTHENTICATION:-


Almost all websites have auth
There are complicated ways (Login with google...) to do auth
Easiest is a username password based auth
facebook
Facebook helps you connect and share with the people in your life.
Email address or phone number
Password
Log in
Forgotten password?
Create new account
Create a Page for a celebrity, brand or business.

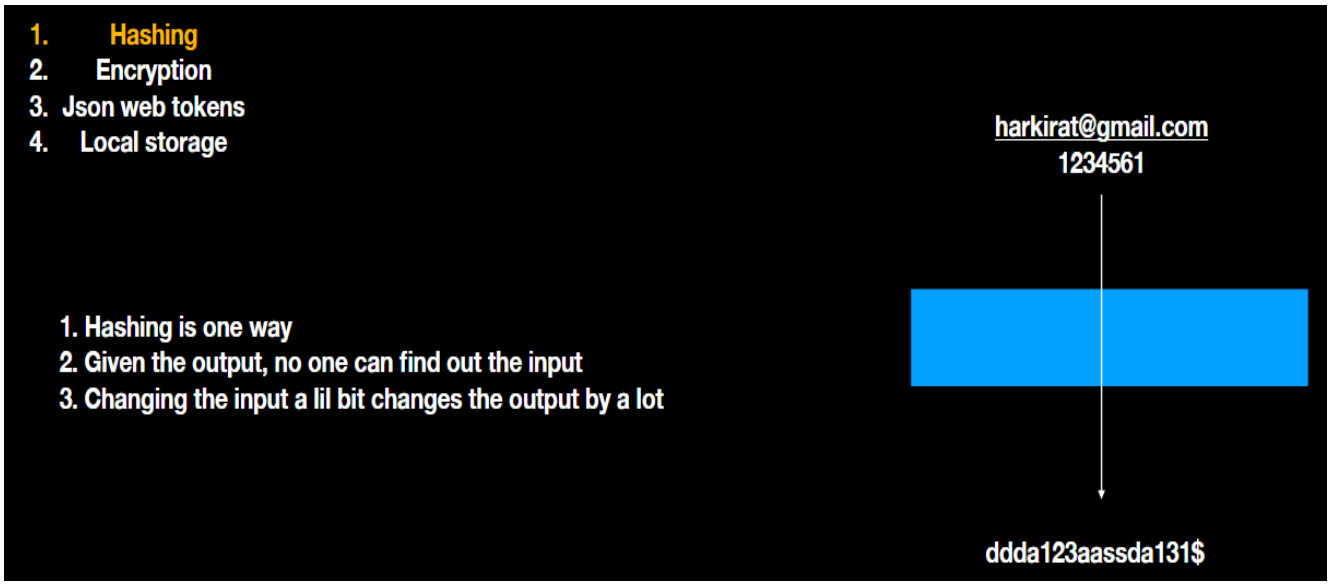

Before we get into authentication
Lets understand some cryptography jargon
1. Hashing
2. Encryption
3. Json web tokens
4. Local storage

1.) Hashing:-

Hashing in authentication refers to the process of converting a plaintext password into a fixed-length string of characters, called a hash, using a cryptographic hash function. This hashed value is then stored in a database or system instead of the original password. When a user attempts to log in, the password they provide is hashed using the same algorithm, and the resulting hash is compared to the stored hash.

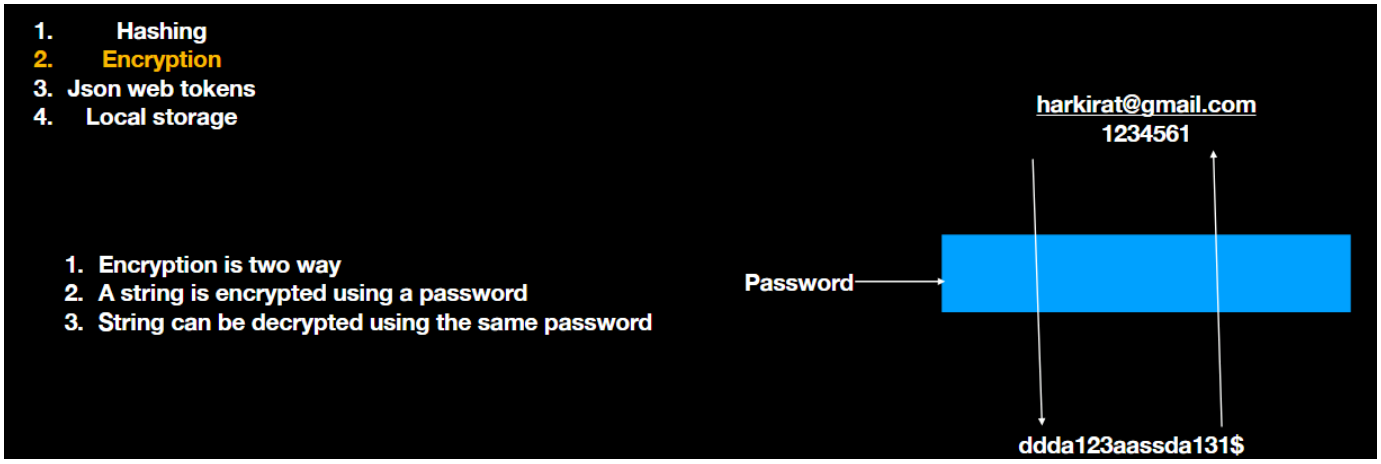
Hashing is a fundamental technique used in authentication for several reasons:

1. **Security**: Hashing helps protect user passwords by storing them in a non-reversible format. Even if an attacker gains access to the database storing hashed passwords, they cannot easily reverse the process to obtain the original passwords. This enhances the security of user accounts and mitigates the impact of data breaches.
2. **Data Integrity**: Hashing ensures data integrity by generating a unique hash value for each input. Even a small change in the input data will result in a significantly different hash value, making it easy to detect any tampering or unauthorized modifications.
3. **Salted Hashes**: To further enhance security, authentication systems often use salted hashes. A salt is a random value added to the password before hashing, resulting in a unique hash even for identical passwords. Salting prevents attackers from using precomputed hash tables (rainbow tables) to crack hashed passwords efficiently.
4. **Password Verification**: During authentication, the user-provided password is hashed using the same algorithm and compared to the stored hash. If the hashes match, it indicates that the user-provided password is correct, and access is granted. If they do not match, access is denied.

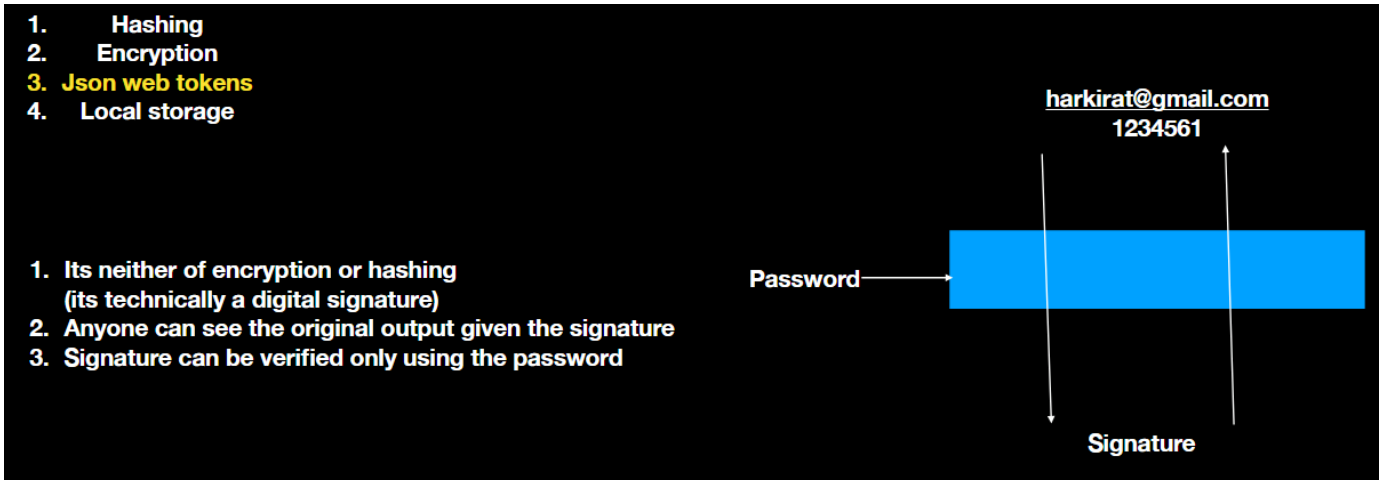


It is a one-way technique, we can convert the input to hash, but we can't convert hash to input.

2.) Encryption:-



3.) JSON Web Tokens:-



It only works for JSON inputs.

It takes the input and generates the token.

4.) Local Storage:-

1. Hashing
2. Encryption
3. Json web tokens
4. Local storage

A place in your browser where you can store some data
Usually things that are stored include -

1. Authentication tokens
2. User language preference
3. User theme preference