AJAX

- AJAX stands for Asynchronous JavaScript And XML.
- AJAX just uses a combination of:
    1. A browser built-in XMLHttpRequest object
    2. JavaScript and HTML DOM

Use of AJAX :

- AJAX enables the developer to reload only a few parts of the web page without affecting or reloading other components.

Send Request using AJAX :

- Using AJAX we can send requests on the server to access data in two ways :
    1. By creating an instance of XMLHttpRequest
    2. By fetch() method

XMLHttpRequest Object :

- The XMLHttpRequest object is used to request data from a server.
- To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object.
- Example:
  xhr.open("GET", "ajax_info.txt", true);
  xh.send();

Method	**Description**
open(*method, URL, async*)	Specifies the type of request *method*: the type of request: GET or POST *url*: the server (file) location *async*: true (asynchronous) or false (synchronous)
send()	Sends the request to the server (used for GET)
send(*string*)	Sends the request to the server (used for POST)

Fetch :

- Fetch is an interface for making an AJAX request in JavaScript. It is implemented widely by modern browsers and is used to call an API.

- Example :

  const promise=fetch(URL,[options])
  where options can be method type, request header field(s), data, etc.

AJAX methods :

AJAX supports methods like GET, POST, PUT, PATCH, DELETE, etc.

GET :
Generally, GET is used to access data from the server.
POST :
Usually, used to send data to server.
PUT :
Used to replace given data with requested data.
PATCH :
Used to update data into server.

JSON data :

- JSON stands for Java Script Object Notaion.
- JSON is a lightweight format for storing and transporting data.
- JSON Syntax Rules
  1. Data is in name/value pairs
  2. Data is separated by commas
  3. Curly braces hold objects
  4. Square brackets hold arrays
- Example :

```
{
        “firstName”:”Krinsi”,
        “details”:[
                {“full-name”:”Krinsi Kayada”,“age” : 20},
                {“city”:”Rajkot”}
        ]
}
```

Serialization :

- Serialization converts an in-memory data structure to a value that can be stored or transferred.
- we can serialize the JSON object by passing it into
  the JSON.stringify() function.

- Example :

```
JSON.stringify(usersObject);
```

Deserialization :

- The conversion from serialized string to an in-memory data structure is deserialization. We can accomplish this with the JavaScript JSON.parse() function.
- Example :

```
const usersObject = JSON.parse(responseFromServer);
```