**Research Assignment**

1. **HTTP and basic working of the web**
2. HTTP stands for hypertext transfer protocol; it is a protocol used on web to exchange information between the server and the client. It uses Request and Responses to perform the necessary actions. The client basically makes and HTTP request to a web server by performing a particular action and the server in response to that request send back a webpage/HTML/Other Files. server does not keep any data (state) between two requests.
3. There are many HTTP methods:
   1. GET = Retrieves data from the server

If you want to check the details of a student, whose entry number is 2021CS1XXXX then you can use this

* 1. POST = Submit data to the server

If you want to insert the data of a new student who has just joined the institute, then in such case POST method is used

* 1. PUT = Updates the data already on the server or create it if it doesn’t exist

If a student changes his/her department then entry number must be updated using this method

* 1. DELETE = Deletes the data on the server

If a student decides to leave the college, then his/her data must be deleted from the server database by using this protocol.

* 1. PATCH = This Method is used to make a partial update to a resource

If a student changes his/her department **(DepC to CS XD)** then entry number must be updated using this method

1. The Headers change and tell us about the requested address, the error code, the method through which the request was sent. It can be seen in Browser (Google Chrome) -> Developer Tools -> Network -> Headers

The **User-Agent** [request header](https://developer.mozilla.org/en-US/docs/Glossary/Request_header) is a featured string that lets servers and network peers identify the application, operating system, and version of the requesting [user agent](https://developer.mozilla.org/en-US/docs/Glossary/User_agent). Also, to get other data about the users and load preferences for it.

* 1. User-Agent request header tool was created initially by Mozilla. The UA string of Firefox itself is broken down into four components:
  2. Example=

Mozilla/5.0(platform;rv:geckoversion)Gecko/geckotrailFirefox/firefoxverioon

* 1. History of it can be read in detail at

<https://webaim.org/blog/user-agent-string-history/>

In brief, basically Netscape and Microsoft remained in constant conflict which eventually lead to micro-soft using Netscape’s name to popularize it’s own technology and making it compatible with Mozilla killing the original technology and then Netscape backfiring by creating Mozilla with Gecko Layout Engine, Soon other browsers which came in later used the higher versions of similar technology since the string initially started with Mozilla/1.22 and so on. Mozilla/5.0 so till date the User-String starts with Mozilla only.

* 1. Gecko is a layout engine created by Mozilla whose basic purpose is to process HTML, CSS, JavaScript sent to a browser and also help API’s to function. Gecko includes, among other things, a networking stack, graphics stack, layout engine, a JavaScript virtual machine, and porting layers.

1. **Web Cookies Work**
2. Cookies are small pieces of text sent to your browser by a website you visit. They help that website remember information about your visit, which can both make it easier to visit the site again and make the site more useful to you. Other technologies, including unique identifiers used to identify a browser, app or device, pixels, and local storage, can also be used for these purposes. Cookies are used for Session Management, Personalization, Tracking.
3. Attributes

Reference:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie>

1. Set-Cookie: <cookie-name>=<cookie-value>
2. Set-Cookie: <cookie-name>=<cookie-value>; Expires=<date>
3. Set-Cookie: <cookie-name>=<cookie-value>; Max-Age=<number>
4. Set-Cookie: <cookie-name>=<cookie-value>; Domain=<domain-value>
5. Set-Cookie: <cookie-name>=<cookie-value>; Path=<path-value>
6. Set-Cookie: <cookie-name>=<cookie-value>; Secure
7. Set-Cookie: <cookie-name>=<cookie-value>; HttpOnly
8. Set-Cookie: <cookie-name>=<cookie-value>; SameSite=Strict
9. Set-Cookie: <cookie-name>=<cookie-value>; SameSite=Lax
10. Set-Cookie: <cookie-name>=<cookie-value>; SameSite=None; Secure
11. // Multiple attributes are also possible, for example:
12. Set-Cookie: <cookie-name>=<cookie-value>; Domain=<domain-value>; Secure; HttpOnly

A cookie typically archives the content name and content value like unique ID for the user or the site name.

1. Google uses cookies for advertising, including serving and rendering ads, personalizing ads or your, limiting the number of times an ad is shown to a user, muting ads you have chosen to stop seeing, and measuring the effectiveness of ads. The ‘NID’ cookie is used to show Google ads in Google services for signed-out users, while the ‘ANID’ and ‘IDE’ cookies are used to show Google ads on non-Google site
2. Cookies can show uncomfortable and unusual ads to the users to market the product of vendors who are willing to pay extra money to sell and marketize their products more. As a user moves around the web, their browser uses the FLoC algorithm to work out its "interest cohort", which will be the same for thousands of browsers with a similar recent browsing history. The browser recalculates its cohort periodically, on the user's device, without sharing individual browsing data with the browser vendor or anyone else. However, people are concerned about the privacy implications of tailored advertising, which currently relies on techniques such as tracking cookies and device fingerprinting which can reveal your browsing history across sites to advertisers or ad platforms. The FLoC proposal aims to allow ad selection in a way that better protects privacy. This is the criticism against it.
3. **Cross-Origin Resources**
4. Usually, the browser makes the request to the server by using the scripts in the web-browser. But some malicious websites might use scripts to access personal information from the user and make it show as if the request was made from the original website.
5. The [Access-Control-Request-Method](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Request-Method) header notifies the server as part of a preflight request that when the actual request is sent, it will do so with a POST request method. The [Access-Control-Request-Headers](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Request-Headers) header notifies the server that when the actual request is sent, it will do so with X-PINGOTHER and Content-Type custom headers. Now the server has an opportunity to determine whether it can accept a request under these conditions.
6. CORS also relies on a mechanism by which browsers make a "preflight" request to the server hosting the cross-origin resource, in order to check that the server will permit the actual request. In that preflight, the browser sends headers that indicate the HTTP method and headers that will be used in the actual request.
7. JWT Tokens
   1. JWT stand for JSON web token
   2. JSON Web Tokens consist of three parts separated by dots (.), which are Header, Payload and Signature. Header contains the information about the token, payload mentions about the user and kind of the access that the user has, and the signature is used to verify the message wasn't changed along the way
8. In case of the JWT approach **the server** does not need to maintain a DB of session Id for lookup.

**ADVANTAGES**

Since userId is got by decrypting the JWT token, no DB call is required to get userId, so somewhat faster that session approach. Servers can be scaled separately, without the need share session DB. This makes the JWT approach a great option for micro-services architecture. The same token can be used to authenticate on different servers (as long as server has the access token secret), without the need to share session DB, this also allows for a completely separate authentication server, that can be solely responsible to issue “access Tokens” and “refresh Tokens”.

**DISADVANTAGES**

Since JWT tokens cannot be “invalidated” (without maintaining them in a shared db.), in JWT approach the logout length precision is set by the expiration length of the access token. However, the “access token” lifespan can be kept short (typically 10 to 15 mins), so that tokens are automatically “invalidated” after the duration. Anti-pattern: Sometimes additional and unnecessary information is stored in the JWT. The JWT token should primarily contain user information, and the data authorized to be accessed by that user should be provisioned and managed as a separate service on that respective server.

1. JWT can not only be used for authentication of both message sender/receiver, but it can also ensure the message integrity as well, leveraging a digital signature hash value of the message body to ensure the message integrity during transmission.

JWTs can be either signed, encrypted or both. If a token is signed, but not encrypted, everyone can read its contents, but when you don't know the private key, you can't change it. Otherwise, the receiver will notice that the signature won't match anymore.