Assignment-1 Course- QA\_trainee

Q.1 How HTTPS works behind the scene?

 Ans: A protocol for secure Internet communication is HTTPS (Secure Hypertext Transfer Protocol). Data transmission between the client (browser) and server is secured and secure with HTTPS. Understanding the technology that makes HTTPS feasible is necessary to comprehend how HTTPS functions.

1. SSL/TLS encryption: The Secure Sockets Layer (SSL) and TLS (Transport Layer Security), which are protocols, offer encryption and authentication for Internet connections.

The SSL/TLS protocol is used to create a secure connection between the client and the server whenever a user sends a request to a server using HTTPS.

2. A digital certificate is a document that uses digital signatures to identify the server. The server's name, public key, and digital signature are all included in the certificate. The server displays its digital certificate to the client during an HTTPS connection, and the client verifies that the certificate is legitimate and issued by a certificate authority (CA).

3. Public Key Infrastructure (PKI): PKI is a network of certificate authorities (CAs), digital certificates, and other infrastructure components that guarantees the security of digital data exchange. Digital certificates that authenticate servers and clients are issued by CAs.

4. Handshake operation: When a client connects to a server using HTTPS, a handshake is carried out between the client and server.

The client and server communicate information about the encryption techniques they both support during the handshake, and the client establishes the server's identity by verifying the appropriate digital certificate. Client and server can begin sending secure data after the handshake is complete.

To summarize, HTTPS uses handshakes to create secure connections and SSL/TLS encryption to protect the connection between the client and server. It also uses digital certificates to authenticate the server and PKI to administer digital certificates.

Q.2 What are different http methods available and what they exactly do.?

Ans: There are a number of HTTP methods available, but these are the ones that are most frequently used:

GET - Data can be retrieved from a server using the GET technique. A server responds to a GET request by providing the requested information, typically in the form of an HTML document or another sort of file.

POST - When creating or updating a resource, data is sent to a server using the POST method. A POST request is sent to a server, which evaluates it and responds with a status code indicating whether the request was successful or unsuccessful.

PUT - A resource on the server can be updated using the PUT technique. A server that receives a PUT request updates the resource with the information in the request body.

DELETE - A resource on the server can be deleted using the DELETE method. A server deletes the requested resource when a DELETE request is made to it.

PATCH - To update a resource with a number of modifications, use the PATCH technique.

Q.3Understand and explain the use of various http response codes.?

Ans: Some of the most typical HTTP response codes and their definitions are as follows:

1xx - Informational Response: These are temporary answers that are used to let the client know that now the server has received and is still processing their request.

100 Continue: The initial portion of the request has been received, and the client should transmit the remaining portion of the request, according to this code.

The server has successfully received, comprehended, and accepted the request if the response code is 2xx.

When a server returns the requested data after successfully processing a request, it displays the code 200 OK.

201 This code denotes that a new resource has been generated on the server after the request was successfully handled.

204 No Content: The request was properly processed by the server, but no data could be delivered.

These response codes, which stand for "3xx Redirection Responses," mean that the resource you sought has been moved to a new URL, and your browser should route you there.

The requested resource has been permanently relocated to a different URL, as indicated by the status code 301.

302 Found: A temporary redirect to a different URL has been made for the requested resource.

304 Not Modified: The resource is still valid in the client's cache, therefore the server does not need to deliver a new copy.

4xx Client Error Responses: These response codes signify that the client's request contained an error.

400 Bad Request: Because the request was invalid, the server was unable to interpret

401 Unauthorized: Before accessing the resource, the client must authenticate themselves.

404 Not Found: The requested resource was not found by the server.

When working with web apps and APIs, it's crucial to comprehend these codes.

Q.4What are the different web communication protocols and their use cases?

Ans:The main protocol for web communication is HTTP (Hypertext Transfer Protocol), which is used to request and send data between web servers and clients. A client sends a request to a server using the request/response protocol known as HTTP, and the server replies with a message containing the data that was requested.

In order to secure data transmission between web servers and clients, HTTPS (Hypertext Transfer Protocol Secure) was developed. Sensitive data, such as financial and personal data, is frequently transmitted using HTTPS.

File transfers between servers and clients take place via the FTP (File Transfer Protocol) protocol. Web developers frequently utilize FTP to upload files to web servers.

Q.5 Pros and cons of Single page and multi page applications.

Ans:

Single-page applications have advantages:

Faster response times: SPAs only load once and do not require page refreshes, which improves user experience and speeds up loading times.

Reduced server demand: SPAs do not necessitate a complete page reload on each request, which lowers server load and improves scalability.

a single-page application's drawbacks:

Initial load time: Due to the necessity of downloading all essential code and files at the start of the session, SPAs may have a longer initial load time.

Complexity: Because SPAs heavily rely on JavaScript, which may be challenging to debug and optimize, they might be more difficult to create and maintain.

Multi-Page Applications' Benefits

Simpler development: Because MPAs depend less on intricate JavaScript code, they may be simpler to create and maintain.

Faster initial load time: Since MPAs only load the assets and code required for each page, they can have a faster initial load time.

MPAs typically work better on older browsers and electronic devices.

The drawbacks of multi-page applications

Slow reaction time: MPAs call for page reloads, which can make the user experience less interactive and cause response times to be slower.

Increased server load: Because MPAs require the server to completely reload the page for each request, they might make scalability issues worse.