

## **1. What is Web 3.0, and how is it different from Web 1.0 and Web 2.0?**

Web 1.0 was the first generation of websites with one-way communication. It was characterized by Simple Banner Ads.. Basic Content Delivery was introduced. Personal Websites were rare. Contents had directories or taxonomy systems. Customers had no interaction with the websites, they could only read the website.

Web 2.0 is all about allowing the user to interact with and contribute content to the website. This is done by providing some means for the user to enter a comment, upload a picture, or “like” something that has been added by someone else for example on Facebook or Twitter and blogging. Web 2.0 is characterized by Automatic texts, and graphical and interactive ads replaced by simple banners. Website, image, and video tagging replaced taxonomy methods.

Web 3.0 is the next generation in the evolutionary progression of the World Wide Web. It is characterized by AI which personalizes ads. Cloud-based file-sharing systems will prevail. The semantic web will take the place of Wikipedia and User-centric live streams will replace blogs.

## **2. What are the functional differences between the front end of a web application and its back end?**

The front end of a website is what the user sees and interacts with. It is also known as the presentation layer. The front end is built using three components HTML, CSS and JavaScript. The front-end developers need to have a good understanding of these languages and they must have a creative mindset and an eye for design.

The back end is part of the website that users don't see. It refers to the server side, which is responsible for returning the page, storing and fetching user data. It is also known as the data access layer. Languages like Python, PHP, or Ruby are often used to build content management systems and large complex web applications. There is interaction between the front and back of a website so both front and back developers work together. Back-end developers need to have more logic than creativity.

## **3. In your own words, explain the process that takes place from when you type a URL into the address bar in your browser until you finally view the page you have requested. Include the HTTP request-response cycle in your answer and provide an example of the HTTP response and request messages.**

When you URL (Uniform Resource Locator) into your browser address bar for example jw.com you want to reach the server where the browser is located. The

browser performs a DNS (Domain Name System) lookup to find the IP address associated with the domain name. The DNS must check different places for the IP address. The DNS then gets converted into an IP address so it can reach the website. This connection ensures reliable data exchange between the browser and the server. The browser then sends an HTTP request to the server specifying what it wants for example a GET request for jw.com. The server receives the request and processes it. It locates the file associated with the requested resource. The server also handles authentication, security checks, and other tasks. The server then sends back an HTTP response. The response includes an HTTP status code indicating success, headers, and the actual content requested. The browser receives the response, parses the HTML content, and fetches additional resources like embedded links, images, CSS, JavaScript files, etc. The web page jw.com will be rendered and the web page will be displayed. All this happens in milliseconds.

The HTTP request-response cycle is a process that occurs when a client communicates with a server over the web. This cycle shows the steps involved from when the client sends a request to the server and when the server responds. The cycle is as follows:

1. The user provides a client with a URL.
2. The client then builds a request for information.
3. The server receives this request and uses it to build a response that contains the requested information.
4. The response is sent back to the client in the requested format to be rendered by the client

Some examples of HTTP request-response are:

#### Client request

GET / jw.htm HTTP/1.1

User\_Agent: Mozilla/4.0

Host: [www.jw.com](http://www.jw.com)

Accept-Language: en-us

Accept-Encoding: gzip, deflate

Connection: Keep Alive

### Server Response

HTTP/1.1 200 OK

Date: Tue, 16 July 2024 12:28:53 GMT

Server: Apache /2.2.14(Win 32)

Last-Modified: Wed 22 May 2024 19:20:34 GMT

Content-Length: 88

Content-Type: text/html

Connection: Closed

<html>

    <body>

        <hi>Jw, World</hi>

    <body>

</html>

### **If jw.com page does not exist:**

#### Client request

GET / jw.html HTTP/1.1

User Agent: Mozilla/4.0

Host: [www.jw.com](http://www.jw.com)

Accept-Language: en-us

Accept-Encoding: gzip, deflate

Connection: Keep Alive

### Server response

HTTP/1.1 404 Not Found

Date: Tue, 16 May 2024 12:28:53 GMT

Server: Apache /2.2.14(Win 32)

Last-Modified: Wed 22 May 2024 19:20:34 GMT

Content-Length: 250

Content-Type: text/html; charset=iso-8859-1

Connection: Closed

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<html>

<head>

    <title>404 Not Found</title>

</head>

<body>

    <h1>Not Found</h1>

    <p>The requested URL /jw.html was not found on this server.</p>

Server. </p>

</body>

</html>