



Web Development Overview

Model Answer

[Visit our website](#)

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

- Web 3.0 represents the next phase of the Internet, characterised by decentralisation, blockchain technology, and an emphasis on user control over data and digital assets.
- Web 1.0 was the initial, static version of the web, offering read-only content with minimal user interaction.
- Web 2.0 brought about dynamic content, interactivity, social media, and user-generated content, but it also led to centralised control by large tech companies.
- Web 3.0 aims to shift control back to users, promoting a more decentralised, secure, and private online experience.

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

- The front end of a web application is the part users directly interact with. It includes the visual design, layout, and user interface (UI). Technologies like HTML, CSS, and JavaScript are typically used for front-end development.
- The back end, on the other hand, operates behind the scenes on the server, managing the database, server logic, and application functionality. It handles data processing, user authentication, and communicates with the front end.

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.

When you enter a URL in your browser, a series of steps occurs before the webpage is displayed. First, the browser performs a **DNS (Domain Name System) lookup**. This converts the human-readable URL into an IP address, identifying the server where the website is hosted. Once the server is located, the browser initiates an **HTTP/HTTPS request**, which follows the request-response cycle fundamental to web interactions.

In this cycle, the browser (client) sends an HTTP request to the server. This request includes various details, such as the requested file location (like `/index.html`), the HTTP method (such as `GET`), and HTTP headers, which contain information about the client (e.g., user-agent, language preferences). For example, a basic HTTP request might look like this:

```
GET /index.html HTTP/1.1
```

Host: www.example.com

User-Agent: Mozilla/5.0

Upon receiving the request, the server processes it and prepares an **HTTP response**. This response consists of a status line indicating the request's success or failure (e.g., **200 OK** for success, **404 Not Found** if the page doesn't exist), headers with additional information (such as **Content-Type** to specify file format), and the requested content itself, typically in HTML. An example response might look like this:

HTTP/1.1 200 OK

Content-Type: text/html; charset=UTF-8

Content-Length: 3054

Once the browser receives this response, it begins the **rendering process**. It interprets the HTML, CSS, JavaScript, and other resources, arranging them to visually display the webpage. As it renders, additional requests may be sent to retrieve linked resources, like images or stylesheets. Together, these steps work seamlessly to deliver the interactive pages we use daily.

This process encapsulates the entire **HTTP request-response cycle**, from the initial request to the server's response and finally to the display of the webpage on your screen.