1.What is client-side and server-side in web development, and what is the main difference between the two?

Client-side, refers to processes that are carried out on the user's device, typically in the user's web browser. These processes are executed after the website or web application has been delivered to the user's device, and they can include tasks such as rendering and displaying a web page, handling user interactions, or running JavaScript code.

Server-side, on the other hand, refers to processes that are carried out on the web server, where the website or web application is hosted. These processes are typically executed by the server before the website or web application is delivered to the user's device, and they can include tasks such as retrieving data from a database, rendering a web page, or handling user input.

The main difference between the two are:

Client-side	Server-side
Client-side code runs on the user's computer	Server-side code runs on the server.
Client-side code is executed by the user's web browser	Server-side code is executed by the web server.
Client-side code is used for creating dynamic user interfaces and providing interactivity,	Server-side code is used for processing user input, accessing databases, and generating dynamic content.
Client-side processes are executed on the user's device after the web application is delivered.	Server-side processes are executed on the web server before the web application is delivered to the user's device.

2.What is an HTTP request and what are the different types of HTTP requests?

HTTP stands for **Hypertext Transfer Protocol**.

HTTP works as a request-response protocol between a client and server.
Example: A client (browser) sends an HTTP request to the server; then the server returns a response to the client.

The different types of HTTP requests are:

GET, POST, PUT, DELETE, PATCH, HEAD, COPY, OPTIONS, LINK, UNLINK, LOCK, CONNECT etc.

3.What is JSON and what is it commonly used for in web development?

JSON stands for **JavaScript Object Notation**. JSON is a **text format** for storing and transporting data. **JSON** is often used hen data is sent from a server to a web page. **JSON** is "self-describing" and easy to understand.

A common use of JSON is to exchange data to/from a web server.

JSON is commonly used in web development for sending and receiving data between client-side and server-side applications. It is often used as an alternative to **XML** (Extensible Markup Language) and other formats because it is less verbose and easier to read and understand.

4.What is a middleware in web development, and give an example of how it can be used?

Middleware is a (loosely defined) term for any software or service that enables the parts of a system to communicate and manage data. It is the software that handles communication between components and input/output, so developers can focus on the specific purpose of their application.

An example is logging middleware, which logs incoming requests and outgoing responses for analysis and debugging purposes. This middleware can capture information such as the request URL, HTTP method, response status, and response time. It can also log any errors that occur during request processing.

5.What is a controller in web development, and what is its role in the MVC architecture?

A controller is the central point of the web application and organizes everything behind the scenes before the users see it.

MVC (Model-View-Controller) is a pattern in software design commonly used to implement user interfaces, data, and controlling logic. It emphasizes a separation between the software's business logic and display.

In summary, the controller is an essential component of the MVC architecture in web development. It helps to separate the concerns of the application, making it easier to manage and maintain over time. By handling requests and coordinating the flow of

data between the model and view, the controller ensures that the application is both responsive and scalable.