

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

Answer-

Client Side Server : Client-side development, sometimes referred to as front-end development, is a type of development that involves programs that run on a client's or user's device. The client-side Server is responsible for presenting the user interface and handling user interactions.

Server Side Development: Server-side development sometimes called back-end development is a type of development that involves programs that run on a server. The Server Side Development handles the processing and storage of data, communication with database as well as generating the web pages to be sent to the client-side. User Don't see this development because it's Happens on server.

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

Answer-

HTTP acronym for HyperText Transfer Protocol is a commonly used internet protocol that allows the server-side and the client-side to communicate with each other. So that's mean HTTP request is a message sent by a client to a server and generate a response for specific action. The HTTP request consists of several components, including a request method, URL, headers, and optional body data.

The different types of HTTP requests are GET, POST, PUT, DELETE, PATCH, HEAD etc. We are most commonly use GET, POST.

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

Answer:

JSON is an acronym for JavaScript Object Notation. It is an open standard format, which is lightweight and text-based, designed explicitly for human-readable data interchange. It is a language-independent data format. JSON is an open standard for exchanging data on the web. It supports data structures like objects and arrays. So, it is easy to write and read data from JSON.

In web development, JSON is often used in the following ways:

Sending data: JSON is widely use to sending data from the server to the client.

Receiving data :JSON is also widely use receiving data from the client to the server.

Data storage :JSON can also be used to store data in a database. For example, a web application might use JSON to store a list of users in a database.

API Communication: JSON is also used to Communicate with APIs. A developer always use JSON to communicate with another developer API's

So JSON is a versatile data format that can be used for a many of tasks when we develop a application.

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

In web development, middleware is a software component that sits between the application and the server or framework. It acts as a bridge that handles and manages various tasks related to incoming requests and outgoing responses such as authentication, logging, error handling and more.



```
1
2 // Create a middleware function that checks if the user is logged in.
3
4 const authMiddleware = (req, res, next) => {
5
6   // Get the user from the request context.
7   const user = req.user;
8
9   // If the user is not logged in, redirect to the login page.
10  if (!user) {
11    res.redirect("/login");
12  } else {
13
14    // The user is logged in, so continue with the request.
15    next();
16  }
17 };
18
19 // Apply the middleware to the routes that require authentication.
20 app.get("/dashboard", authMiddleware, (req, res) => {
21
22   // The user is logged in, so show them the dashboard.
23   res.render("dashboard");
24 });
```

In this example, the authMiddleware function checks if the user is logged in. If the user is not logged in, the middleware redirects the user to the login page. If the user is logged in, the middleware continues with the request and renders the dashboard view.

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

Answer:

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

The role of a controller in the MVC architecture that is Handling user requests then interacting with the model and

preparing data for the view. So, the controller plays a crucial role in achieving this separation by coordinating the flow of data and actions between the view and the model.