Ostad

Assignment

on

Module Seven

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

- **SERVER-SIDE**

  Server-side 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.

- **CLIENT-SIDE**

  Client-side, on the other hand, 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.

- **MAIN DIFFERNCE BETWEN THE TOW?**

  - Server-side and client-side refer to the location where certain tasks or processes are carried out in a web application.
  - Server-side processes are executed on the web server before the web application is delivered to the user's device.
  - Client-side processes are executed on the user's device after the web application is delivered.
  - Server-side processes have more access to resources and are more secure, while client-side processes have less access to resources and are potentially less secure.

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

According to IBM, An HTTP request is made by a client, to a named host, which is located on a server. The aim of the request is to access a resource on the server.

❑ **DIFFERENT TYPE OF HTTP REQUEST.**

HTTP (Hypertext Transfer Protocol) specifies a collection of request methods to specify what action is to be performed on a particular resource. The most commonly used HTTP request methods are GET, POST, PUT, PATCH, and DELETE. These are equivalent to the CRUD operations (create, read, update, and delete).

GET: GET request is used to read/retrieve data from a web server. GET returns an HTTP status code of 200 (OK) if the data is successfully retrieved from the server.

POST: POST request is used to send data (file, form data, etc.) to the server. On successful creation, it returns an HTTP status code of 201.

PUT: A PUT request is used to modify the data on the server. It replaces the entire content at a particular location with data that is passed in the body payload. If there are no resources that match the request, it will generate one.

PATCH: PATCH is similar to PUT request, but the only difference is, it modifies a part of the data. It will only replace the content that you want to update.

DELETE: A DELETE request is used to delete the data on the server at a specified location.

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

❑ **ACCORDING TO W3SCHOOLS.COM JSON IS**

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is a lightweight data-interchange format
- JSON is plain text written in JavaScript object notation
- JSON is used to send data between computers
- JSON is language independent

Commonly used for in web development

JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax. It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa).

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

Middleware is software that different applications use to communicate with each other. It provides functionality to connect applications intelligently and efficiently so that you can innovate faster. Middleware acts as a bridge between diverse technologies, tools, and databases so that you can integrate them seamlessly into a single system. The single system then provides a unified service to its users. For example, a Windows frontend application sends and receives data from a Linux backend server, but the application users are unaware of the difference.

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

- Controller

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. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.