

## Part 1: Short Answer Questions

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

Answer: Client-side is responsible for user interface rendering and interaction. Server-side handles business logic, data processing, and communication with databases and external services. The main difference between the two is where from it runs.

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

Answer: HTTP request is Hypertext Transfer Protocol request.

Different types of HTTP requests is given below:

GET: The GET method is used to retrieve data from a server.

POST: The POST method is used to submit data to be processed by the server.

PUT: The PUT method is used to update an existing resource on the server.

PATCH: The PATCH method is used to make partial updates to an existing resource.

DELETE: The DELETE method is used to remove a specified resource from the server.

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

Answer: JSON is JavaScript Object Notation.

Structure, Readability, Data Interchange are the reasons that's why it is used commonly in web development.

Structure: JSON is a text-based format consisting of key-value pairs.

Readability: JSON is designed to be human-readable and easy to understand.

Data Interchange: JSON is commonly used as a format for data interchange between a client and a server.

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

Answer: Middleware works between the client and the server and handle various tasks during the request-response cycle. When server request is sent to the specific route before going to controlled the data request limiting, securing task are done by middleware where it working like middle man.

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

Answer: Controller is a component or a module that handles the incoming requests from the client.

In MVC architecture controller acts as an intermediary between the model and the view. Receive and handle the incoming requests from the client. Interpret and validate the user's input data.