**Part A ( Theory )**

**1. Mention the working of Internet Website in Terms of Front-end & Back-end Divisions.**

Front-end & Back end are the two main divisions of an internet website.

The Front-end, also known as the client-side, refers to the visible and interactive components of a website that users interact with directly. It encompasses the design, layout, and user interface of the website. The front-end division involves the following elements:

a. HTML (Hypertext Markup Language): HTML provides the structure and semantic meaning to the content on the website. It defines the elements, such as headings, paragraphs, images, links, forms, and more.

b. CSS (Cascading Style Sheets): CSS is responsible for styling the HTML elements. It controls the visual presentation of the website, including colors, fonts, layout, and positioning. CSS allows developers to customize the appearance of various elements on the page.

c. JavaScript: JavaScript is a programming language that adds interactivity and dynamic functionality to the website. It enables actions like form validation, animations, real-time updates, and handling user interactions. JavaScript frameworks/libraries like React, Angular, or Vue.js are often used to simplify complex front-end development tasks.

The Back-end, also known as the server-side, is responsible for the behind-the-scenes operations that make a website function. It involves the server, database, and application logic. The back-end division consists of

a. Server: The server handles incoming requests from users' browsers and processes them. It can be built using various server-side technologies such as Node.js, Ruby on Rails, Django, or PHP. The server interacts with the database, executes the application logic, and generates responses to send back to the client.

b. Database: The database stores and manages the website's data. It can be a relational database (e.g., MySQL, PostgreSQL) or a NoSQL database (e.g., MongoDB, Firebase). The back-end interacts with the database to retrieve or update data based on user requests.

c. Application Logic: The application logic includes the code and algorithms that define how the website functions and processes user input. It handles tasks like user authentication, data validation, business logic implementation, and integration with external services or APIs.

**2) What are tags in HTML? Explain the each category of tag with an Example?**

Tags in HTML are predefined elements that define the structure and content of a web page. They are enclosed in angle brackets (<>) and usually come in pairs, with an opening tag and a closing tag. Each category of tag serves a specific purpose and helps organize and format the content of a web page.

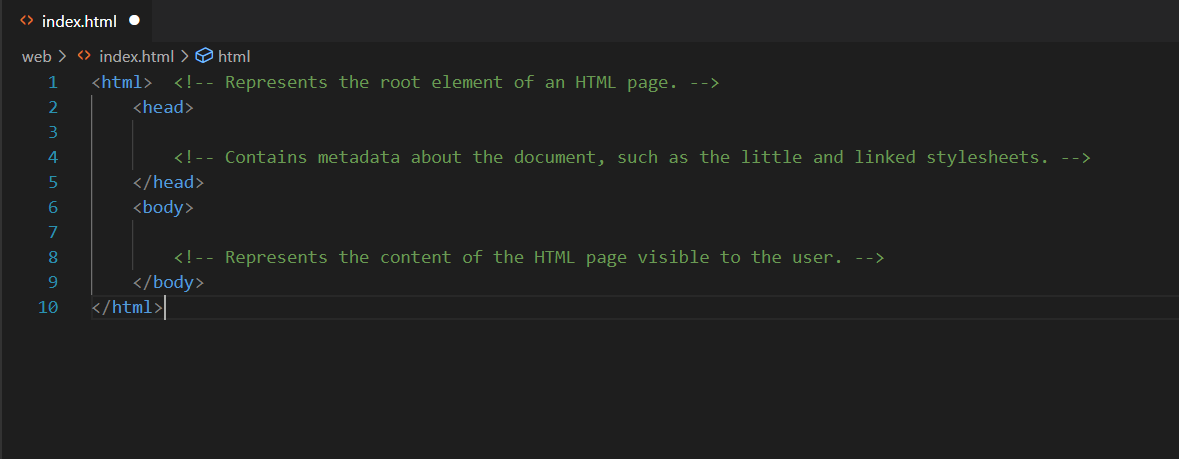
Examples:

A) Structural Tags:

**<html> :** Represents the root element of an HTML page.

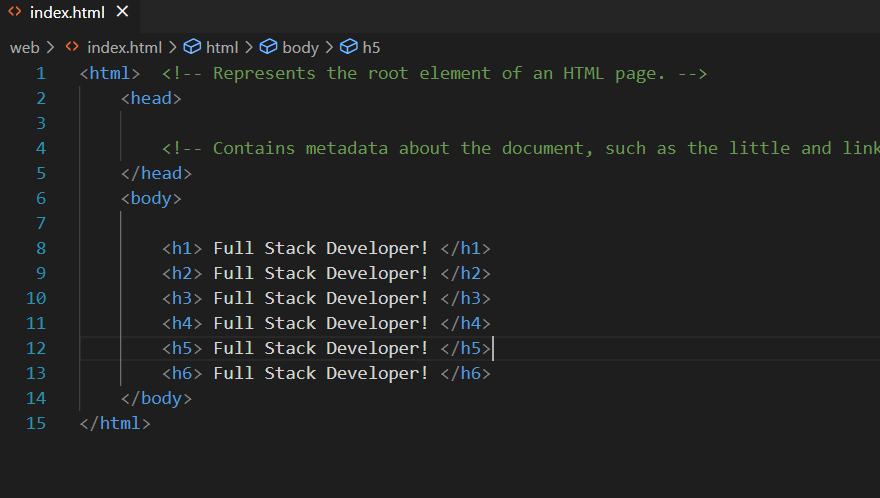
**<head>:** Contains metadata about the document, such as the little and linked stylesheets.

**<body>:** Represents the content of the HTML page visible to the user.



B) Heading Tags:

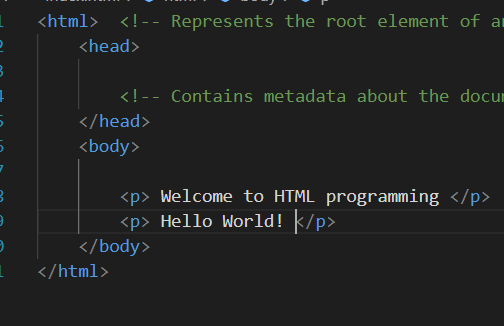
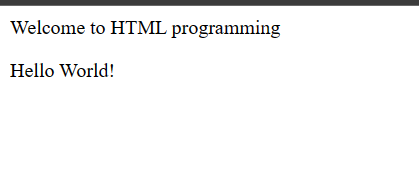
<h1> to <h6>: Represents different levels of headings, with <h1> being the highest (most important) and <h6> being the lowest.



RESULTS



C) Paragraph Tags: It represents a paragraph of a text, can be of any text.

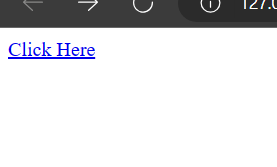
 

D) Link Tags:

<a>: Creates a hyperlink to another web page or a specific location within the same page. It requires an href attribute to specify the target URL.

Example: <a href="https://www.Google.com">Click here</a>

When you click on “Click here” it directly redirects to google search.



E) Image Tags:

<img>: Inserts an image into the web page. It requires a src attribute to specify the image source (URL) and an alt attribute to provide alternative text for accessibility.

Example: <img src="image.jpg" alt="give any name to the image">



F) List Tags:

<ul>: Represents an unordered (bulleted) list.

<ol>: Represents an ordered (numbered) list.

<li>: Represents an individual list item within <ul> or <ol>.

G) Division Tags:

<div>: Defines a division or section in an HTML document. It is often used as a container for other HTML elements, allowing for easier styling and layout.

H) Form Tags:

<form>: Creates an HTML form for user input. It encloses form elements such as input fields, checkboxes, radio buttons, and submit buttons.

<input>: Represents an input field within a form. It can have various types (text, email, password, etc.) specified by the type attribute.

By using the appropriate tags, we can create well-structured and semantically meaningful web pages.

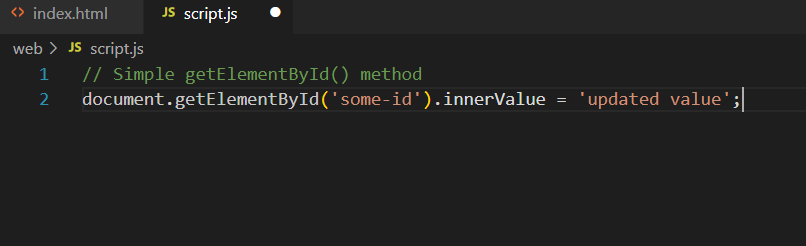
**3. Explain the working Procedure of Virtual DOM.**

DOM stands for ‘Document Object Model’. In simple terms, it is a structured representation of the HTML elements that are present in a webpage or web-app. DOM represents the entire UI of your application. The DOM is represented as a tree data structure. It contains a node for each UI element present in the web document. It is very useful as it allows web developers to modify content through JavaScript, also it being in structured format helps a lot as we can choose specific targets and all the code becomes much easier to work with.

While sir explaining on Day4 we might have seen him making use of ‘getElementById()’ or ‘getElementsByClassName()’ method to modify the content of the DOM. Every time there is a change in the state of your application, the DOM gets updated to reflect that change in the UI.

As DOM is represented as a tree itself, updating the tree here is not a costly operation indeed we have a lot of algorithms on trees to make the updates fast. What’s proving to be costly is every time the DOM gets updated, the updated element and its children have to be rendered again to update the UI of our page. Like this each time there is a component update, the DOM needs to be updated and the UI components have to be re-rendered.

Example:



When we are writing that code, in browser

1. The browser parses the HTML to find the node with this id.
2. It removes the child element of this specific element.
3. Updates the element (DOM) with the ‘updated value’.
4. Recalculates the CSS for the parent and child nodes.
5. Update the layout.
6. Finally, traverse the tree and paint it on the screen(browser) display.

So as we know now that updating the DOM not only involves changing the content, it has a lot more attached to it. Also recalculating the CSS and changing the layouts involves complex algorithms, and they do affect the performance. So React has a different approach to dealing with this, as it makes use of something known as Virtual DOM.

React uses Virtual DOM exists which is like a lightweight copy of the actual DOM(a virtual representation of the DOM). So for every object that exists in the original DOM, there is an object for that in React Virtual DOM. It is exactly the same, but it does not have the power to directly change the layout of the document. Manipulating DOM is slow, but manipulating Virtual DOM is fast as nothing gets drawn on the screen. So each time there is a change in the state of our application, the virtual DOM gets updated first instead of the real DOM.

**4) Mention some Differences between MySQL and No SQL**

MySQL and NoSQL (Not Only SQL) are two different types of database management systems with distinct characteristics. Here are some differences between MySQL, a popular relational database management system, and NoSQL databases.

1. Data Model:

MySQL: MySQL follows a structured and predefined schema with a fixed table structure, where data is organized into tables with rows and columns. It enforces data integrity and supports relationships between tables through foreign keys.

NoSQL: NoSQL databases offer flexible and schema-less data models. They can handle unstructured, semi-structured, or polymorphic data formats. NoSQL databases typically use key-value, document, columnar, or graph-based data models.

2. Scalability:

MySQL: MySQL traditionally scales vertically by adding more resources to a single server. It can handle large datasets but may face limitations in terms of scalability and high availability.

NoSQL: NoSQL databases are designed to scale horizontally by distributing data across multiple servers or clusters. They excel in handling large-scale applications and can scale easily to accommodate growing datasets and user loads.

3. Query Language:

MySQL: MySQL uses Structured Query Language (SQL) as its primary query language. SQL provides a standardized way to interact with the database and perform operations like querying, inserting, updating, and deleting data.

NoSQL: NoSQL databases may have different query languages or APIs depending on the type of database. Some NoSQL databases provide SQL-like query languages, while others use custom query languages or APIs specific to their data models.

4. Schema Flexibility:

MySQL: MySQL has a predefined schema, which means the structure of the database and tables needs to be defined before storing data. Any changes to the schema often require modifying the table structure.

NoSQL: NoSQL databases offer schema flexibility, allowing developers to store and retrieve data without a predefined schema. This flexibility allows for agile development and handling dynamic data structures.

5. ACID Transactions:

MySQL: MySQL supports ACID (Atomicity, Consistency, Isolation, Durability) transactions, ensuring data integrity and reliability in a multi-user environment.

NoSQL: NoSQL databases may provide varying degrees of ACID transaction support. Some NoSQL databases prioritize high scalability and availability over strict ACID guarantees, providing eventual consistency or relaxed consistency models.

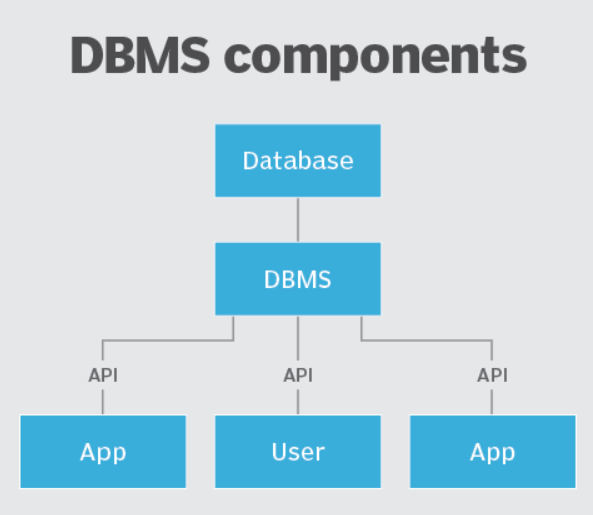
6. Use Cases:

MySQL: MySQL is widely used for traditional relational database applications that require complex queries, transactions, and data consistency. It is often chosen for applications that involve structured data and well-defined relationships between entities.

NoSQL: NoSQL databases excel in scenarios that involve large-scale data storage, high-speed data ingestion, flexible data models, and horizontal scalability. They are commonly used in web applications, real-time analytics, content management systems, and scenarios where rapid development and scalability are essential.

**5) Explain any one DBMS Technology in your own words.**

A database management system (DBMS) is system software for creating and managing databases. A DBMS makes it possible for end users to create, protect, read, update and delete data in a database. The most prevalent type of data management platform, the DBMS essentially serves as an interface between databases and users or application programs, ensuring that data is consistently organized and remains easily accessible.



One popular DBMS (Database Management System) technology is MongoDB. MongoDB is a NoSQL database that provides a flexible and scalable solution for storing and retrieving data. Here's an explanation of MongoDB in my own words:

MongoDB is a document-oriented database that stores data in a flexible and schema-less manner. It uses a JSON-like data format called BSON (Binary JSON) to represent and organize data. Instead of storing data in tables with fixed schemas like traditional relational databases, MongoDB stores data in collections of documents.

A document in MongoDB is a self-contained data structure that can hold a wide variety of data types and sizes. It is similar to a row in a table but with more flexibility. Documents can have nested structures, allowing for rich and complex data representations. This makes MongoDB well-suited for handling semi-structured and unstructured data.

One of the key features of MongoDB is its ability to scale horizontally by distributing data across multiple servers or clusters. It supports automatic sharding, which means that as the data grows, MongoDB can distribute and balance the data across multiple nodes, allowing for high scalability and performance.

MongoDB also provides powerful querying capabilities. It has a flexible query language that allows for complex queries and supports various operators and indexes to optimize query performance. MongoDB's queries can search based on specific fields, perform aggregations, and utilize advanced features like text search and geospatial queries.

In addition, MongoDB offers built-in support for replication and high availability. It can replicate data across multiple servers to ensure data durability and fault tolerance. Replication provides redundancy and allows for failover in case of server failures.

With its flexible data model, scalability, and powerful querying capabilities, MongoDB is well-suited for a wide range of use cases. It is commonly used in web applications, content management systems, real-time analytics, IoT (Internet of Things) applications, and any scenario that requires flexibility, scalability, and high-performance data storage and retrieval.