1. Difference between HTML Webpage & React Webpage

Aspect	HTML Webpage	React Webpage
Structure	Built using static HTML, CSS, JS files.	Built using components (JSX + JS) that render HTML dynamically.
Rendering	Browser loads and renders full HTML each time.	Uses **Virtual DOM** — only updates parts that change.
Reusability	HTML elements are static; no modular reusability.	React components are **reusable and modular**.
Interactivity	Needs JavaScript manually for dynamic actions.	React handles interactivity with **state & props**.
Performance	Slower when reloading pages.	Faster updates — **Single Page Application (SPA)**.
Data Handling	No built-in data management.	Handles dynamic data easily using **state management**.

- HTML → Static & simple.
- React → Dynamic, component-based, SPA framework.

2. Difference between Angular & React

Feature	Angular	React
Type	Full-fledged **Framework**	**Library** for UI
Language	Uses **TypeScript** by default	Uses **JavaScript/JSX** (optional TypeScript support)
Learning Curve	Steeper — has many built-in features	Easier — focuses mainly on UI
Data Binding	**Two-way binding**	**One-way binding**
DOM Handling	Real DOM	Virtual DOM
Architecture	MVC (Model-View-Controller)	Component-based
Size	Larger bundle size	Lightweight
Developed By	Google	Meta (Facebook)

- Angular = Complete ecosystem (routing, forms, services).
- React = UI library; flexible with third-party tools.

3. What is TypeScript?

TypeScript is a **superset of JavaScript** that adds **static typing**.

Example:

```
let name: string = "Mani";
let age: number = 25;
```

If you write `age = "twenty"`, TypeScript will throw an **error**.

Benefits:

- Type safety (detects errors early)
- Better autocompletion & IntelliSense
- Easier debugging
- Used in frameworks like **Angular** and **React (TSX)**

4. Key Concepts

Term	Meaning & Example
Variable	Container for storing data. `let name = "Lunar";`
Declaration	When you declare variable: `let x;`
Initialization	Assign value: `x = 10;`
Lexical Scope	Variable access depends on where it's **declared in code structure**, not where it's called.
Block Scope	Variable available only inside `{}`. `let` and `const` follow block scope.
Function Scope	Variables declared inside function accessible only within that function. (`var` uses function scope).
Hoisting	JS moves **declarations** (not initializations) to top of scope. Example: You can use a `var` before declaring it.
Closure	A function that **remembers variables** from its outer scope even after that scope is gone. Example:

```
function outer() {
  let count = 0;
  return function inner() {
    count++;
    console.log(count);
  };
}
const counter = outer();
counter(); // 1
counter(); // 2
```

5. Why Node.js is used in Web Development?

Node.js allows **JavaScript to run on the server-side**, not just in browsers.

Reasons it's used:

- Handles **backend logic**, APIs, and databases.
- Enables **full-stack JavaScript** (React frontend + Node backend).
- Fast — built on **V8 engine** (Google Chrome's JS engine).
- Supports **real-time apps** (like chat, live updates).

✅ In short:
Node = Backend environment for JavaScript.

6. Which came first — Node, Angular, or React?

Technology	Released Year	Created By
Node.js	**2009**	Ryan Dahl
AngularJS	**2010**	Google
React	**2013**	Meta (Facebook)

Order:
Node.js → Angular → React