□ HTML5 (20 Questions)

1. **Q:** What is HTML5?
   A: HTML5 is the latest version of HTML used to structure web content with new elements, attributes, and APIs.
2. **Q:** Name any 3 new elements in HTML5.
   A: `<header>`, `<footer>`, `<section>`.
3. **Q:** What is the use of `<canvas>` in HTML5?
   A: It is used to draw graphics via JavaScript.
4. **Q:** What is the difference between `<div>` and `<section>`?
   A: `<div>` is generic; `<section>` is for semantically grouped content.
5. **Q:** What is localStorage?
   A: It stores data in the browser that persists even after refresh.
6. **Q:** What is the difference between `localStorage` and `sessionStorage`?
   A: `localStorage` persists until manually cleared; `sessionStorage` clears when tab is closed.
7. **Q:** What is the use of `<video>` tag?
   A: Embeds video content directly in a web page.
8. **Q:** What is semantic HTML?
   A: HTML that clearly describes its meaning to browser and developers.
9. **Q:** Can we use multimedia in HTML5?
   A: Yes, using `<audio>` and `<video>` tags.
10. **Q:** What is the use of the `placeholder` attribute?
    A: It shows a short hint in input fields.
11. **Q:** What is the `required` attribute?
    A: Makes a form field mandatory.
12. **Q:** What are `data-*` attributes?
    A: Custom data attributes for JavaScript access.
13. **Q:** What is the purpose of `<article>` tag?
    A: Represents independent, self-contained content.
14. **Q:** What does `<nav>` tag define?
    A: It defines navigation links.
15. **Q:** What is the use of `<progress>` tag?
    A: Shows progress of a task.
16. **Q:** Can we validate forms in HTML5?
    A: Yes, using built-in input types and attributes.
17. **Q:** What are input types in HTML5?
    A: Examples: `email`, `date`, `range`, `color`.
18. **Q:** What is a `<details>` tag?
    A: Used to show/hide additional info.
19. **Q:** What is a `<summary>` tag?
    A: Defines a summary for `<details>`.
20. **Q:** What is a responsive web design?
    A: Design that works on all screen sizes.

□ JavaScript (20 Questions)

21. **Q:** What is JavaScript?
    A: A scripting language used to make web pages interactive.
22. **Q:** What is the difference between `var`, `let`, and `const`?
    A: `var` is function-scoped, `let` and `const` are block-scoped. `const` cannot be reassigned.
23. **Q:** What is a function in JavaScript?
    A: A reusable block of code.
24. **Q:** What are arrow functions?
    A: A shorter syntax for writing functions: `()=>{}`.
25. **Q:** What is an array?
    A: A collection of items stored in a single variable.
26. **Q:** What are objects in JS?
    A: Collections of key-value pairs.
27. **Q:** What is DOM?
    A: Document Object Model – it represents the web page.
28. **Q:** How do you select elements in JS?
    A: Using `document.getElementById`, `querySelector`, etc.
29. **Q:** What is `eventListener`?
    A: It listens for events like click, input, etc.
30. **Q:** What is the difference between == and ===?
    A: == compares value; === compares value and type.
31. **Q:** What is `NaN`?
    A: Not a Number.
32. **Q:** What is `undefined` in JS?
    A: A variable declared but not assigned.
33. **Q:** What is `null`?
    A: Intentional absence of value.
34. **Q:** What is `typeof`?
    A: Returns the data type of a variable.
35. **Q:** What is `setTimeout()`?
    A: Delays code execution.
36. **Q:** What is a callback function?
    A: A function passed as an argument to another function.
37. **Q:** What is hoisting?
    A: Variables and functions are moved to the top of scope.
38. **Q:** What is a promise?
    A: Object representing the eventual result of an async operation.
39. **Q:** What is async/await?
    A: Syntax to handle promises more easily.
40. **Q:** What are JavaScript data types?
    A: String, Number, Boolean, Object, Undefined, Null, Symbol, BigInt.

□ Node.js (20 Questions)

41. **Q:** What is Node.js?
    A: A runtime environment for executing JS on the server.
42. **Q:** Is Node.js single-threaded?
    A: Yes, but it uses asynchronous non-blocking I/O.
43. **Q:** What is `npm`?
    A: Node Package Manager – installs libraries.
44. **Q:** What is a package.json file?
    A: Contains metadata about a Node project.
45. **Q:** How do you install a package?
    A: `npm install package-name`.
46. **Q:** What is a module?
    A: Reusable file or library.
47. **Q:** How do you import a module in Node.js?
    A: `require('module')`.
48. **Q:** What is the use of `fs` module?
    A: For file operations like read/write.
49. **Q:** What is the use of `http` module?
    A: To create a web server.
50. **Q:** How do you create a server in Node.js?
    A: Using `http.createServer()`.
51. **Q:** What is middleware in Node.js?
    A: Functions that handle request/response.
52. **Q:** What is event-driven programming in Node.js?
    A: Executes code on certain events.
53. **Q:** What is the use of `process` object?
    A: Gives info about current Node process.
54. **Q:** What is `global` object in Node.js?
    A: Global namespace object, like `window` in browsers.
55. **Q:** What is a callback in Node.js?
    A: Function executed after another completes.
56. **Q:** What is the purpose of `exports`?
    A: To expose functions/variables from a module.
57. **Q:** What is `Buffer` in Node.js?
    A: For handling binary data.
58. **Q:** What is `stream` in Node.js?
    A: For handling streaming data (e.g., reading files).
59. **Q:** Can Node.js access the file system?
    A: Yes, using `fs` module.
60. **Q:** What is `cluster` module in Node.js?
    A: For running multiple Node processes.

□ Express.js (20 Questions)

61. **Q:** What is Express.js?
    A: A web framework for Node.js.

62. **Q:** How do you install Express?
    A: `npm install express`.
63. **Q:** How do you create an Express app?
    A: `const app = express();`
64. **Q:** What is middleware in Express?
    A: Code that runs before reaching the route.
65. **Q:** What is a route in Express?
    A: Defines how an app responds to HTTP requests.
66. **Q:** How do you define a GET route?
    A: `app.get('/path', callback);`
67. **Q:** How do you define a POST route?
    A: `app.post('/path', callback);`
68. **Q:** What is `req` and `res`?
    A: `req` is request object; `res` is response object.
69. **Q:** What is `app.listen()` used for?
    A: Starts the server.
70. **Q:** What is `body-parser`?
    A: Middleware to parse request body.
71. **Q:** How do you send JSON in response?
    A: `res.json({ key: 'value' });`
72. **Q:** What is routing in Express?
    A: Mapping URLs to handlers.
73. **Q:** How to handle 404 in Express?
    A: Add a middleware at the end with `res.status(404).send('Not found');`
74. **Q:** What is `next()` in Express middleware?
    A: Passes control to next middleware.
75. **Q:** How do you handle errors in Express?
    A: Using error-handling middleware with 4 arguments.
76. **Q:** Can you serve static files using Express?
    A: Yes, using `express.static()`.
77. **Q:** What is `cors` in Express?
    A: Middleware to enable Cross-Origin Resource Sharing.
78. **Q:** How to parse URL parameters in Express?
    A: Using `req.params`.
79. **Q:** How to parse query string in Express?
    A: Using `req.query`.
80. **Q:** How to return HTML in Express?
    A: `res.send('<h1>Hello</h1>');`

□ TypeScript (20 Questions)

81. **Q:** What is TypeScript?
    A: A superset of JavaScript with static typing.
82. **Q:** How do you declare a variable with type in TS?
    A: `let name: string = "Manasa";`

83. **Q:** What is the benefit of using TypeScript?
    A: Type safety and better tooling.
84. **Q:** What are types in TypeScript?
    A: `string`, `number`, `boolean`, `any`, `void`, etc.
85. **Q:** What is a tuple in TypeScript?
    A: An array with fixed types and length.
86. **Q:** What is an interface in TS?
    A: Describes shape of an object.
87. **Q:** What is a union type?
    A: Variable can hold multiple types: `string | number`.
88. **Q:** How do you compile TS to JS?
    A: `tsc filename.ts`
89. **Q:** What is `any` type?
    A: A flexible type that disables type checking.
90. **Q:** What is the use of `readonly` in TS?
    A: Makes a property unchangeable.
91. **Q:** What is `enum` in TypeScript?
    A: A way to define named constants.
92. **Q:** What are type aliases?
    A: Custom names for types: `type ID = number | string`.
93. **Q:** What is optional parameter?
    A: Parameter that may or may not be provided: `function(x?: string)`
94. **Q:** Can we use classes in TS?
    A: Yes, with proper type annotations.
95. **Q:** What is `never` type?
    A: For functions that never return.
96. **Q:** What is `void` type?
    A: For functions that return nothing.
97. **Q:** Can you use `this` keyword in TS?
    A: Yes, same as JavaScript.
98. **Q:** Can we use TypeScript in Node.js?
    A: Yes, with `ts-node` and `tsc`.
99. **Q:** What is the difference between interface and type in TS?
    A: Both are similar; interfaces can be extended more flexibly.
100. **Q:** Is TypeScript object-oriented?
    A: Yes, it supports classes, interfaces, inheritance.