HTML

1. What is HTML?

HTML stands for HyperText Markup Language. It is the standard markup language (define the structure of document) used to create the structure and content of web pages.

2. Advantages of HTML5?

5 advantages are:

- **New semantic elements** - <header> <footer> <nav> <section> <article>.
- **New form input types** - <input type=”data/email”>.
- **Audio and video support** - <audio> <video>.
- Mobile compatibility.
- Simpler code.

3. What’s the purpose of DOCTYPE?

The DOCTYPE declaration specifies the version of HTML the document is using and helps browsers to render the content correctly.

4. Advantages of <title> tag?

4 advantages are:

- Displayed in the tab of the browser.
- Search engine uses as main heading for search result.
- Default name for bookmark.
- Default title in shared post.

5. What are empty elements?

Elements that doesn’t need content between opening and closing tags. It is also known as self-closing tags or void elements. Example: <img> <input> <br> <hr>.

6. What are semantic elements?

Elements that carry meaning about the structure and content of the web page, providing additional information to both browser and developer. Example: <header> <footer> <nav> <main> <article> <section>.

7. What's the difference between <div> and <span>?

<div> is a block-level element used for grouping and applying styles, while <span> is an inline-level element used for applying styles to small portions of text.

NODE JS

8. What's Node JS?

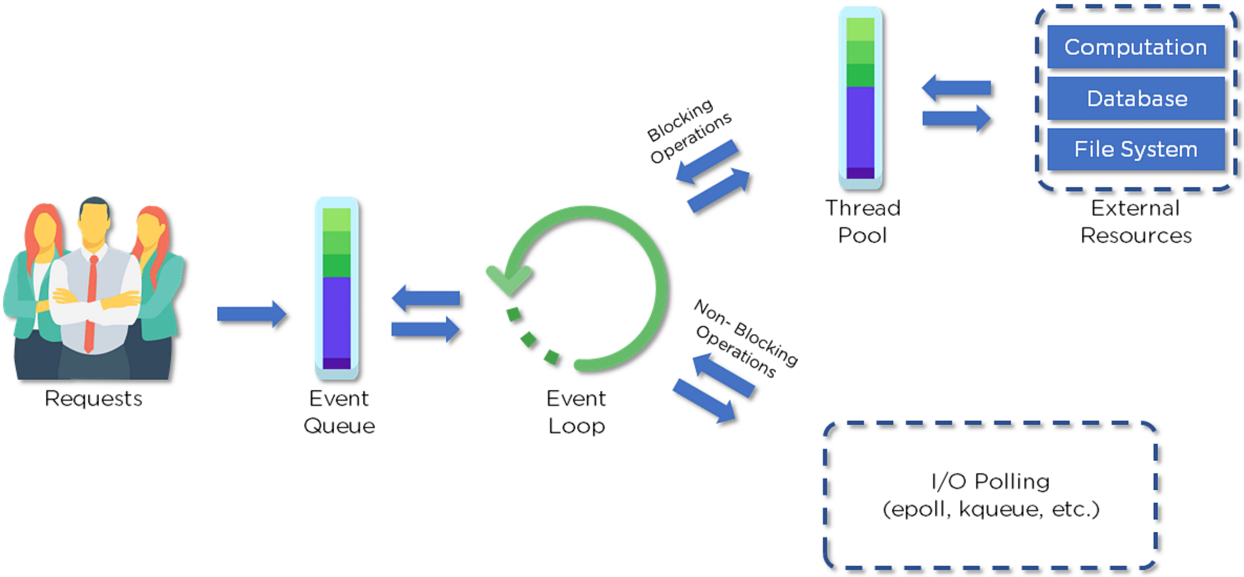
Node JS is an open-source, cross-platform JavaScript runtime environment that allows developers to execute JavaScript code on the server side.

9. Why Node JS?

The Key characteristics are:

- **Server-side development** – Used for building web servers and API's.
- **Asynchronous programming** – Using an event driven architecture.
- **Single programming language** – Use JS for both client and server side.
- **NPM** – Powerful package manager.
- **Real-time application** – Well suited for chat app, online gaming and etc.
- **Development tools** – It has popular frameworks like Express JS, Socket.IO and etc.

10. How Node JS works?



- Client send requests to web server to interact with the web application where that request can be blocking or non-blocking.
- Node JS retrieves the request and add it to Event Queue.
- Then requests are passed one-by-one through the Event Loop where it processes and respond to client as per the response.
- If the request is complex or blocking operation like accessing external resource or database, then a single thread from Thread Pool is assigned and responsible for completing it till Event Loop.
- Finally, Event Loop sends those responses back to the client.

11. Why Node JS is single-threaded?

Node JS is single-threaded by design and this design choice is closely related to it's event-driven, non-blocking I/O model. The key reasons are:

- **Asynchronous, non-blocking I/O**: Optimized for handling large number of concurrent connections and I/O operations.
- **Scalability**: Node JS can scale to support thousands of concurrent connections with lower memory consumption.
- **Simplicity and easy of development**: Avoids issues like thread safety, synchronization and race conditions.

12. What's NPM?

NPM (Node Package Manager) is the default package manager for Node.js. It allows developers to install, manage, and share JavaScript packages and libraries.

13. What are modules in Node JS?

Modules are JS libraries that can be used in our application to include a set of functions by using the require() function with the module's name in it. Example: HTTP, URL, FS and etc.

14. What's the purpose of package.json file?

Heart of our application. Simply, a configuration file that contains metadata about a Node JS project, such as its dependencies, scripts, and other project-specific details.

15. What's callback hell?

Callback hell refers to the nesting of multiple callbacks, leading to unreadable and difficult-to-maintain code. It can be mitigated by using named functions, modularization, and employing Promises or async/await.

16. How Node JS handle errors?

Node JS uses callbacks with the first parameter being reserved for an error object. Developers are expected to check for errors and handle them appropriately.

17. What’s the purpose of process.nextTick function?

It’s used to schedule a callback function to be executed in the next iteration of the event loop. It is often used to defer the execution of a callback to allow the current operation to complete.

18. What’s streams?

Objects that allow reading or writing data continuously. They provide an efficient way to handle large amounts of data without loading the entire dataset into memory.

19. What’s Passport?

Passport is a popular authentication middleware for Node.js. It provides a simple and modular way to implement authentication in Node.js applications. Passport supports many authentication mechanisms, including username/password, social logins like Facebook and Google, and JSON Web Tokens (JWT’s).

EXPRESS JS

20. What’s Express JS?

Express JS is a web application framework for Node JS that simplifies the process of building robust and scalable web applications. It provides a set of features and tools for routing, middleware, and handling HTTP requests and responses.

21. What’s middleware?

Middleware functions in Express JS have access to the request, response, and the next function. They can modify the request or response objects, end the request-response cycle, and call the next middleware in the stack. It allows various tasks like authentication, logging, error handling, and more, in a modular and reusable way.

22. How routing works?

Express JS uses a routing mechanism to define how an application responds to a client request. Routes are defined using HTTP methods and URL patterns, and each route can have a handler function.

23. Why app.use()?

Is used to mount middleware functions in the Express JS application.

24. What’s the difference between req.params and req.query?

Params contains route parameters extracted from the URL (/users/:userId). Query contains query parameters from the query string in the URL (/products?productId=78).

25. How form submissions are handled?

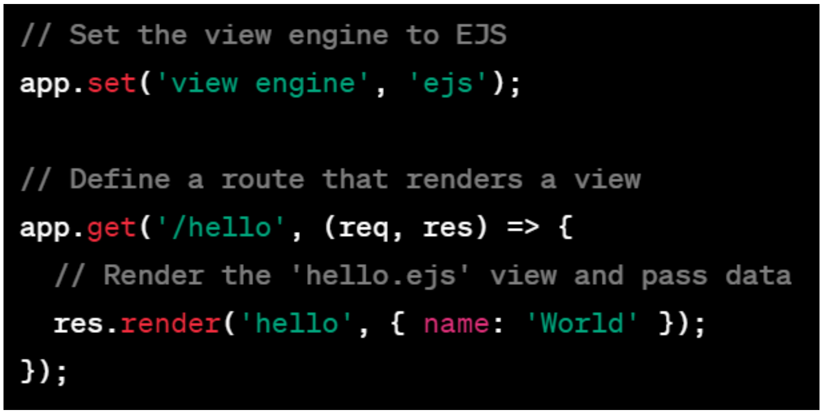
Form submissions can be handled using the body-parser middleware to parse the form data. The data is then accessible through req.body in the route handler.

26. How Express JS handles errors?

Express.js uses middleware functions to handle errors. Error-handling middleware takes four arguments (err, req, res, next) and is defined with an extra parameter, making it recognizable as an error-handling middleware.

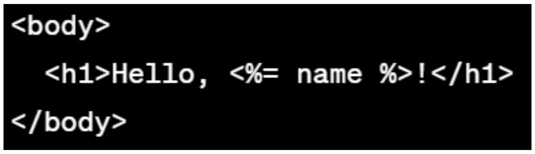
27. What’s template engine?

A template engine in Express.js allows you to dynamically render HTML using templates. Common template engines include EJS, Pug (formerly Jade), and Handlebars.



```
// Set the view engine to EJS
app.set('view engine', 'ejs');

// Define a route that renders a view
app.get('/hello', (req, res) => {
  // Render the 'hello.ejs' view and pass data
  res.render('hello', { name: 'World' });
});
```



```
<body>
  <h1>Hello, <%= name %>!</h1>
</body>
```

28. What’s the purpose of express.static?

Used to serve static files, such as images, CSS, and JavaScript, directly by specifying a directory containing these files.

29. How can we prevent Cross-site scripting (XSS) attacks?

There are different ways to do it:

- Use a Template Engine with Auto-Escape enabled by default.
- Use sanitize-html library.
- Use encodeURIComponent() to validate and encrypt user input.

MONGO DB

30. Explain Mongo DB?

The key features are:

- Is a popular open-source NoSQL document-oriented database that stores data in flexible, JSON-like documents and provides high performance, availability and scalability also it can handle large amount of unstructured or semi-structured data.
- Mongo Shell – CLI tool to interact with Mongo DB instances.
- BSON (Binary JSON) is a binary-encoded serialization of JSON-like documents used in MongoDB. It extends JSON to include additional data types and supports efficient binary encoding and decoding.
- In MongoDB, a collection is a group of MongoDB documents. It is the equivalent of a table in a relational database.
- MongoDB uses the find() method to query documents in a collection.
- The _id field is a default field in every MongoDB document that serves as a unique identifier for that document within a collection.
- A compound index in MongoDB involves creating an index on multiple fields. It allows for more efficient querying when filtering on multiple criteria.
- Sharding is the process of distributing data across multiple machines to improve horizontal scalability. It allows MongoDB to handle large amounts of data and traffic by distributing the data set across multiple servers.
- Namespace - database_name.collection_name.
- Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a higher-level, schema-based abstraction over the raw MongoDB driver, making it easier to work with MongoDB databases in a Node.js environment. Mongoose helps developers define schema, data models, apply validation, and interact with MongoDB using a more convenient and expressive API.