Mern Stack

Q.1. What is the MERN Stack?

MERN Stack is a collection of 4 technologies (MongoDB | Express.js | React.js | Node.js). This technology stack enables a developer to build a web app quickly. One of the primary functions of this stack is a virtual DOM to improve the performance of a web app.

Q.2. What is the role of MongoDB in MERN?

MongoDB is a popular open-source document database, and it can store data in JSON-like documents to properly store complex data. MongoDB has another advantage in that it has an in-built support for cloud environments. With the help of MongoDB, a developer can modify or even sometimes remove the document properties within a collection.

Q.3. How Does React.js Work?

ReactJS has a declarative JavaScript framework to create a dynamic and it allows a developer to create dynamic client-side applications in HTML. This JavaScript library helps a developer build eye-catching user interfaces and has a component-based architecture to encourage code reusability.

Q.4. What are React Hooks?

Hooks is a newly launched feature in React and it allows a developer to use state and other features without any need to write a class. If follows two rules:

– Only Call Hooks at the Top Level

– Only Call Hooks from React Functions

Q.5. What is Callback Hell?

Callback hell is a condition in which a development nests multiple callbacks within a specific function. Here, the shape of the deformed code resembles a pyramid, therefore, it is also known as the “Pyramid of the Doom” because it makes it difficult for a developer to properly maintain the code and understand it.

Q.6. What are the advantages of React.js?

Here are some of the best advantages of React.js that a developer enjoys.

React.js is much easier to learn, maintain, and use

Create dynamic apps with less coding

Provide multiple reusable components

Offer a scope where the developer can test the codes

Comes with strong community support

React.js is rich in terms of both packages & extensions

React has amazing JavaScript support

Q.7. What’s the purpose of Express.js?

Expres.js is a web app development framework to support the Node.js projects. It is one of the best open-source frameworks that manages the flow between the front end and the database.

Q.8. What is a React Key?

In React.js, a key is one of the unique identifiers and is useful for a developer to learn which items are modified or deleted from the lists.

Q.9. What is the concept of Virtual DOM in React.js?

Virtual DOM (VDOM), a programming concept, is the representation of actual DOM and React uses it to update & render its components by minimizing the direct manipulation of the actual DOM.

Q.10. What is JSX in React?

JSX (JavaScript Extension) is a React extension that is famous for allowing developers to write JavaScript codes that appear like HTML codes.

Q.1 How does React work

React creates a virtual DOM. When state changes in a component it firstly runs a "diffing" algorithm, which identifies what has changed in the virtual DOM. The second step is reconciliation, where it updates the DOM with the results of diff.

Q.2 What is props in React

Props are inputs to a React component. They are single values or objects containing a set of values that are passed to React Components on creation using a naming convention similar to HTML-tag attributes. i.e, They are data passed down from a parent component to a child component. The primary purpose of props in React is to provide following component functionality: Pass custom data to your React component, Trigger state changes and Use via this.props.reactProp inside component's render() method.

Q.3 What Is Replication In MongoDB

Replication is the process of synchronizing data across multiple servers. Replication provides redundancy and increases data availability. With multiple copies of data on different database servers, replication protects a database from the loss of a single server. Replication also allows you to recover from hardware failure and service interruptions.

Q.4 What are Higher-Order components

A higher-order component (HOC) is a function that takes a component and returns a new component. Basically, it's a pattern that is derived from React's compositional nature We call them as "pure' components" because they can accept any dynamically provided child component but they won't modify or copy any behavior from their input components.

Q.5 What do you mean by Asynchronous API

All APIs of Node.js library are aynchronous that is non-blocking. It essentially means a Node.js based server never waits for a API to return data. Server moves to next API after calling it and a notification mechanism of Events of Node.js helps server to get response from the previous API call.

Q.6 What is Callback Hell

The asynchronous function requires callbacks as a return parameter. When multiple asynchronous functions are chained together then callback hell situation comes up.

Q.7 What is Reconciliation

When a component's props or state change, React decides whether an actual DOM update is necessary by comparing the newly returned element with the previously rendered one. When they are not equal, React will update the DOM. This process is called reconciliation.

Q.8 Does Mongodb Support Foreign Key Constraints

No. MongoDB does not support such relationships. The database does not apply any constraints to the system (i.e.: foreign key constraints), so there are no "cascading deletes" or "cascading updates". Basically, in a NoSQL database it is up to you to decide how to organise the data and its relations if there are any.

Q.9 How Node prevents blocking code

By providing callback function. Callback function gets called whenever corresponding event triggered.

Q.10 How can you achieve transaction and locking in MongoDB

To achieve concepts of transaction and locking in MongoDB, we can use the nesting of documents, also called embedded (or sub) documents. MongoDB supports atomic operations within a single document.

Q.11 How does Node.js handle child threads

Node.js, in its essence, is a single thread process. It does not expose child threads and thread management methods to the developer. Technically, Node.js does spawn child threads for certain tasks such as asynchronous I/O, but these run behind the scenes and do not execute any application JavaScript code, nor block the main event loop. If threading support is desired in a Node.js application, there are tools available to enable it, such as the ChildProcess module.

Q.12 How to avoid Callback Hell in Node.js

Node.js internally uses a single-threaded event loop to process queued events. But this approach may lead to blocking the entire process if there is a task running longer than expected. Node.js addresses this problem by incorporating callbacks also known as higher-order functions. So whenever a long-running process finishes its execution, it triggers the callback associated. Sometimes, it could lead to complex and unreadable code. More the no. of callbacks, longer the chain of returning callbacks would be. There are four solutions which can address the callback hell problem: Make your program modular, Use async/await mechanism, Use promises mechanism and Use generators.

Q.13 If Node.js is single threaded then how it handles concurrency

Node provides a single thread to programmers so that code can be written easily and without bottleneck. Node internally uses multiple POSIX threads for various I/O operations such as File,

DNS, Network calls etc. When Node gets I/O request it creates or uses a thread to perform that I/O operation and once the operation is done, it pushes the result to the event queue. On each such event, event loop runs and checks the queue and if the execution stack of Node is empty then it adds the queue result to execution stack.

Q.14 What are Pure Components

PureComponent is exactly the same as Component except that it handles the shouldComponentUpdate method for you. When props or state changes, PureComponent will do a shallow comparison on both props and state. Component, on the other hand, won't compare current props and state to next out of the box. Thus, the component will re-render by default whenever shouldComponentUpdate is called.

Q.15 What are React Hooks

Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class. With Hooks, you can extract stateful logic from a component so it can be tested independently and reused. Hooks allow you to reuse stateful logic without changing your component hierarchy. This makes it easy to share Hooks among many components or with the community.

Q.16 What is Aggregation in MongoDB

Aggregations operations process data records and return computed results. Aggregation operations group values from multiple documents together, and can perform a variety of operations on the grouped data to return a single result. MongoDB provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function and single purpose aggregation methods and commands.

Q.17 What is JSX

JSX is a syntax extension to JavaScript and comes with the full power of JavaScript. JSX produces React elements. You can embed any JavaScript expression in JSX by wrapping it in curly braces. After compilation, JSX expressions become regular JavaScript objects. This means that you can use JSX inside of if statements and for loops, assign it to variables, accept it as arguments, and return it from functions:

Q.18 What is ReactDOM

It's a top-level React API to render a React element into the DOM, via the ReactDOM.render method.

Q.19 What is Sharding in MongoDB

Sharding is a method for storing data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

Q.20 What is Stream and what are types of Streams available in Node.js

Streams are a collection of data that might not be available all at once and don't have to fit in memory. Streams provide chunks of data in a continuous manner. It is useful to read a large set of data and process it. There is four fundamental type of streams: Readable, Writeable, Duplex and Transform

Q.21 What is prop drilling

When building a React application, there is often the need for a deeply nested component to use data provided by another component that is much higher in the hierarchy. The simplest approach is to simply pass a prop from each component to the next in the hierarchy from the source component to the deeply nested component. This is called prop drilling.

Q.22 What is Key

A key is a special string attribute you need to include when creating lists of elements. Keys help React identify which items have changed, are added, or are removed.

Q.23 What is a Blocking Code

If application has to wait for some 1/0 operation in order to complete its execution any further then the code responsible for waiting is known as blocking code

Q.24 What is the difference between ShadowDOM and VirtualDOM

Virtual DOM is about avoiding unnecessary changes to the DOM, which are expensive performance-wise, because changes to the DOM usually cause re-rendering of the page. Virtual DOM also allows to collect several changes to be applied at once, so not every single change causes a re-render, but instead re-rendering only happens once after a set of changes was applied to the DOM. Shadow DOM is mostly about encapsulation of the implementation. A single custom element can implement more-or-less complex logic combined with more-or-less complex DOM. An entire web application of arbitrary complexity can be added to a page by an import and but also simpler reusable and composable components can be implemented as custom elements where the internal representation is hidden in the shadow DOM like.

Q.25 What's the Event Loop

The event loop is what allows Node.js to perform non-blocking I/O operations despite the fact that JavaScript is single-threaded by offloading operations to the system kernel whenever

possible. Every I/O requires a callback - once they are done they are pushed onto the event loop for execution. Since most modern kernels are multi-threaded, they can handle multiple operations executing in the background. When one of these operations completes, the kernel tells Node.js so that the appropriate callback may be added to the poll queue to eventually be executed.

Q.26 What's the difference between a "smart" component and a "dumb" component

Smart components manage their state or in a Redux environment are connected to the Redux store. Dumb components are driven completely by their props passed in from their parent and maintain no state of their own.

Q.27 What is Mongoose

Mongoose is an Object Document Mapper (ODM), which means that by using Mongoose, you can define objects with a strongly-typed schema that can be further mapped to a MongoDB document. It offers a schema-based solution for modeling application data. Mongoose comes with built-in typecasting, validation, query building, business logic hooks, and many more out-of-the-box features.

Q.28 What is REPL In Node.Js

REPL or "Read Eval Print Loop" is a simple program that can accept commands, evaluate them, and prints the results. What REPL does is to create an environment that is similar to a Unix/Linux shell or a Window console, wherein you can enter command and system, and it will respond with the output. Here are the functions that REPL performs: READ (This reads the input provided by the user, parses it into JavaScript data structure, and stores it in the memory.), EVAL (This executes the data structure), PRINT (This prints the outcome generated after evaluating the command.) and LOOP (This loops the above command until the user presses Ctrl+C twice.)

Q.29 How to check if an object is an array or not in JavaScript

The best way to find whether an object is instance of a particular class or not using toString method from Object.prototype

Q.30 List down the two arguments that async.queue takes as input in Node.js

Task Function and Concurrency Value