MERN Stack Assignment 2

1. What is MongoDB replication?

Ans.

Replication is basically 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. It also allows recovery from hardware failure and service interruption.

MongoDB achieves replication with the use of a replica set.

A replica set is a group of mongod instances that host the same data set (minimum 3 are required)

In replica, one node is the primary node that receives all write operations. All other instances such as secondaries, apply operations from the primary so that they have the same data set.

Replica set can have only one primary node.

At the time of automatic failover or maintenance, election is established for primary and a new primary node is elected.

After the recovery of the failed node, it again joins the replica set and works as a secondary node.

2. What are some of NodeJs' features?

Ans.

- Node Js library is very fast in code execution.
- Node Js application never buffers any data. These applications simply output the data in chunks.
- Node Js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable.
- Node Js is asynchronous and Event Driven - All APIs of Node Js Library are asynchronous, that is non-blocking. It means Node js based server never waits for API to return data. The server moves to the next API after calling it and the notification mechanism of Event of Node js helps the server to get a response from the previous API call.

3. What is MongoDB's database type?

Ans.

MongoDB is a cross platform, document oriented database that provides high performance, high availability and easy scalability.

MongoDB is classified as a NoSQL database.

MongoDB works on the concept of collection and document.

MongoDb uses JSON-like document having dynamic schema (document in same collection don’t need to have same set of fields, document may hold different types of data).

4. How does Node prevent code from being blocked?

Ans.

Node prevents blocking of code by using a single-threaded event loop.

Node uses a single thread, which means one task can be executed at a time. This is done using stack.

While reading the code from top to bottom, each instruction is pushed into a stack and when its execution is completed, it pops out of the stack.

When it comes across any instruction/function that will take a longer time to execute and can result in delay in popping the stack and execution of further statements. That's where Node Js allow use of Event Loop.

Each time when it encounters such a situation, the process causing delay is offloaded from the stack and the execution of that process continues parallel to further execution of main code. Thus, the callback for that function is pushed into a task queue and the code continues to execute asynchronously. When the process completes its execution, the callback function returns the desired output from that process and resumes normal execution.

5. What exactly do you mean when you say "pure components"?

Ans.

In React Pure Component is basically the component which does not re-render when the value of state and props have been updated with the same value.

If the value of the previous state and props and the new state and props is same, then the component is not re-rendered.

In React Pure Component the state and Props are shallow compared. It is the same as component except that Pure Component takes care of shouldComponentUpdate by itself. If the previous state and props data is same as the next props or state, the component is not Re-rendered.

React provides the PureComponent base class for class components. Class components that extend the React.PureComponent class are treated as pure components.