1.What is MongoDB replication?

a)

MongoDB replication is the process of creating a copy of the same data set in more than one MongoDB server. This can be achieved by using a Replica Set. A replica set is a group of MongoDB instances that maintain the same data set and pertain to any mongod process.

2. What are some of NodeJs features?

a)

i)Asynchronous and Event Driven − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

ii)Very Fast − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

iii)Single Threaded but Highly Scalable − 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 as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

iv)No Buffering − Node.js applications never buffer any data. These applications simply output the data in chunks.

v)License − Node.js is released under the MIT license

3. What is MongoDB's database type?

a)

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

MongoDB provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

4. How does Node prevent code from being blocked?

a)

Blocking operations refer to the pieces of code that block the execution of other code until they are completed. While non-blocking operations allow further pieces of code to execute without making them wait and use callbacks when they are completed.

Thus, blocking code can be said to work synchronously while non-blocking code works asynchronously. The callback is a function that is invoked when a process completes its execution and wants to continue its normal execution with the outer function.

Node.js provides a set of asynchronous I/O primitives in its standard libray that prevent Javascript code from blocking other additional javascript code. In general, Node.js libraries are written using an asynchronous, non-blocking paradigms.

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

a)

React components let us split the UI into independent, reusable pieces, and think about each piece in isolation. React components can be defined by extending from React.Component Class or React.PureComponent Class.