**React.Js Questions**

1. **What is Props?**

**Answer: Props is short form of properties, it is used to send data from parent component to child components**

1. **What are the different phases of React component's lifecycle?**

**Answer: Mouting phase, Updating Phase , Unmounting Phase**

1. **How do you create an event in React?**

**Answer: Same like javascript example in jsx there is button you can onClick event**

1. **What is the primary difference between the ES6 and ES5 standards?**

**Answer: You can write Import and Export in ES6 in ES5 use the older module**

**ES6 there is template Literals (`${Xyz}`) ES6 we can write class**

1. **How do you update the state of a component?**

Answer: Using the usestate hook we can also update the state using redux

1. **What is PureComponent?**

**Answer: Those components are avoiding unnecessary renders this called the PureComponetns**

1. **What is React.cloneElement?**
2. **What is the difference between mapStateToProps() and mapDispatchToProps() ? Give an example.**

1. **How to add multiple middlewares to Redux?**

**Answer:using applyMiddleware**

1. **What is wrong with this example, and how would you go -**

|  |
| --- |
| class App extends React.Component {  constructor(props) {  super(props);  this.state = {  name: this.props.name || 'Anonymous'  }  } render() {  return (  <p>Hello {this.state.name}</p>  );  }  } |

**Answer:Remove this from name:props.name**

1. **Can you explain the differences between all those ways. What happens when you click each of the buttons?**

|  |
| --- |
| class App extends React.Component {  constructor() {  super();  this.name = 'MyComponent';  this.handleClick2 = this.handleClick1.bind(this);  }  handleClick1() {  alert(this.name);  }  handleClick3 = () => alert(this.name);render() {  return (  <div>  <button onClick={this.handleClick1()}>click 1</button>  <button onClick={this.handleClick1}>click 2</button>  <button onClick={this.handleClick2}>click 3</button>  <button onClick={this.handleClick3}>click 4</button>  </div>  );  }  } |

Answer:

Button Click 1 handleCllick1 function immediately invoked (not able to respond)

Button Click 2 when clicked the function is excuted

Button Click 3 it will display the alert message my componets

Button Click 4 it will also display the alert message my componets

1. **What's the issue with this component. Why? How would you go about fixing it?**

|  |
| --- |
| class App extends React.Component {state = { search: '' }handleChange = event => {/\*\*  \* This is a simple implementation of a "debounce" function,  \* which will queue an expression to be called in 250ms and  \* cancel any pending queued expressions. This way we can  \* delay the call 250ms after the user has stoped typing.  \*/  clearTimeout(this.timeout);  this.timeout = setTimeout(() => {  this.setState({  search: event.target.value  })  }, 250);  }render() {  return (  <div>  <input type="text" onChange={this.handleChange} />  {this.state.search ? <p>Search for: {this.state.search}</p> : null}  </div>  )  }  } |

1. **Why is props passed to the super() function in React?**

**Answer: Because super function is used to invoked the constructor of the parent class and it ensures that the props object is properly initialized.**

1. **What are the benefits of HOC?**
2. **Why do we need a Router to React?**
3. **Give an example of class component and functional component.**

**Answer:**

**Class Components**

**import React, { Component } from 'react';**

**class ClassComponent extends Component {**

**constructor(props) {**

**super(props);**

**this.state = {**

**count: 0**

**};**

**}**

**incrementCount = () => {**

**this.setState({ count: this.state.count + 1 });**

**};**

**render() {**

**return (**

**<div>**

**<h2>Class Component</h2>**

**<p>Count: {this.state.count}</p>**

**<button onClick={this.incrementCount}>Increment</button>**

**</div>**

**);**

**}**

**}**

**export default ClassComponent;**

**Functional Components**

**import React, { useState } from 'react';**

**function FunctionalComponent() {**

**const [count, setCount] = useState(0);**

**const incrementCount = () => {**

**setCount(count + 1);**

**};**

**return (**

**<div>**

**<h2>Functional Component</h2>**

**<p>Count: {count}</p>**

**<button onClick={incrementCount}>Increment</button>**

**</div>**

**);**

**}**

**export default FunctionalComponent;**

1. **Explain the purpose of “render()” in React. Give an example.**

**Render method is a fundamental part of class -based components**

**Example:**

**import React, { Component } from 'react';**

**class MyComponent extends Component {**

**render() {**

**return (**

**<div>**

**<h1>Hello, {this.props.name}!</h1>**

**<p>{this.props.message}</p>**

**</div>**

**);**

**}**

**}**

**export default MyComponent;**

**Node.Js Questions**

1. **What's the Event Loop?**
2. **What are the advantages of using promises instead of callbacks?**

**Answer:clear Readability,createChain,Error Handing,async flow**

1. **How do you create a simple server in Node.js that returns Hello World?**

Answer: const express = require("express");

const app = express();

const port = 4000;

app.get("/", (req, res) => {

  res.send("Hello, Express!");

});

  res.json({ message: `User created: ${name} (${email})` });

});

// Error handler for 404 Not Found

app.use((req, res) => {

  res.status(404).json({ error: "Not Found" });

});

app.listen(port, () => {

  console.log(`Server running at http://localhost:${port}`);

});

1. **Which of the following statements is valid to import a module in file?**
2. **var fs = require("fs");**
3. var fs = import("fs");
4. package fs;
5. import fs;
6. **How to make node modules available externally?**
7. module.expose
8. module.spread
9. **module.export**
10. **Which of the following are Node.js stream types?**
11. Duplex
12. Readable
13. Writable
14. **All of the above**
15. **What is the default scope of Node.js application?**
16. Global
17. Local
18. Public
19. Private
20. **Which module is used to serve static files in Node.js?**
21. static
22. node-static
23. http
24. node-http
25. **REPL stands for.**
26. Research Eval Program Learn
27. Read Eval Print Loop
28. Read Earn Point Learn
29. Read Eval Point Loop
30. **Which of the following code print the platform of operating system?**
31. console.log('platform : ' + os.platform);
32. **console.log('platform : ' + os.platform());**
33. console.log('platform : ' + os.getPlatform());
34. None of the above.
35. **What is a blocking code?**
36. **What are global objects in Node.js?**
37. **What is the use of EventEmitter in Node.js?**
38. **What is control flow function?**

**MongoDB**

1 What makes MongoDB the best?

Flexibal,jsondata,scalable,richQuery

2 How to do transactions/locking in MongoDB?

3 When and to what extent does data get extended to multi-slice?

4 Compare MongoDB with Couchbase and CouchbaseDB.

5 When do we use a namespace in MongoDB?

6 If you remove an object attribute, is it deleted from the database?

7 How can we move an old file into the moveChunk directory?

8 Explain the situation when an index does not fit into RAM.

9 How does MongoDB provide consistency?

10 Why is MongoDB not chosen for a 32-bit system?

11 How does Journaling work in MongoDB?

12 How can you isolate the cursors from intervening with the write operations?

13 Define MongoDB.

14 Explain the replica set.

15 What are the key features of MongoDB?

16 What is CRUD?

Create Readd Update Delete

17 What is Sharding?

18 What is Aggregation in MongoDB?

19 Define Namespace in MongoDB.

Set - 1

1 How many Triggers are possible in MySQL?

2 What are the types of relationships used in MySQL?

3 How to join three tables in MySQL?

4 What is the default port for MySQL server?

5 Can you tell what are the different set operations available in MySQL?

6 What are the different tables present in Mysql?

7 How can you test for NULL values in a database?

8 What is the difference between the NVL function, IFNULL function, and the ISNULL function?

9 What is self referencing foreign key? Give an example.

10 What is the difference between TRUNCATE and DELETE?