

React, Node.js, MongoDB, JavaScript, and DSA Question Set - 1 (Answer Key)

Question-1

What are React lifecycle methods? List and briefly explain the different lifecycle methods used in React components.

Answer:

React lifecycle methods are special methods in class-based components that get called at different stages of a component's existence: Mounting, Updating, and Unmounting.

- Mounting: constructor(), componentDidMount()
- Updating: shouldComponentUpdate(), componentDidUpdate()
- Unmounting: componentWillUnmount()
- Error Handling: componentDidCatch(), getDerivedStateFromError()

In functional components, lifecycle is handled using Hooks like useEffect().

Question-2

Create a simple counter app using React and useState.

Answer:

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

function Counter() {
  const [count, setCount] = useState(0);
  return (
    <div>
      <h1>Count: {count}</h1>
      <button onClick={() => setCount(count + 1)}>Increment</button>
      <button onClick={() => setCount(count - 1)}>Decrement</button>
      <button onClick={() => setCount(0)}>Reset</button>
    </div>
  );
}

export default Counter;
```

Question-3

What is a closure function in JavaScript?

React, Node.js, MongoDB, JavaScript, and DSA Question Set - 1 (Answer Key)

Answer:

A closure is a function that remembers and accesses variables from its outer scope even after the outer function has executed.

Example:

```
function outer() {  
  let count = 0;  
  return function inner() {  
    count++;  
    console.log(count);  
  }  
}  
  
const counter = outer();  
counter(); // 1  
counter(); // 2
```

Question-4

How to optimize React application performance?

Answer:

- React.memo(), useMemo(), useCallback(), React.lazy(), Suspense
- Avoid unnecessary renders, optimize images, use keys properly, virtualize large lists.

Question-5

What is fs module in Node.js?

Answer:

fs (File System) module allows working with the file system.

Example:

```
const fs = require('fs');  
  
fs.writeFileSync('example.txt', 'Hello World');  
  
const data = fs.readFileSync('example.txt', 'utf8');  
  
console.log(data);
```

React, Node.js, MongoDB, JavaScript, and DSA Question Set - 1 (Answer Key)

Question-6

How to make a navigation bar sticky to top?

Answer:

Use CSS property position: sticky; top: 0;

Question-7

What is MongoDB Aggregation Pipeline?

Answer:

Pipeline processes data in stages like \$match, \$group.

Example:

```
db.orders.aggregate([
  { $match: { status: 'delivered' } },
  { $group: { _id: '$customerId', total: { $sum: '$amount' } } }
]);
```

Question-8

Write MongoDB query to sort by age descending.

Answer:

```
db.students.find().sort({ age: -1 });
```

Question-9

Check if two strings are anagrams.

Answer:

```
function isAnagram(str1, str2) {
  if (str1.length !== str2.length) return false;
  const count = {};
  for (let char of str1) count[char] = (count[char] || 0) + 1;
  for (let char of str2) {
    if (!count[char]) return false;
    count[char]--;
  }
}
```

React, Node.js, MongoDB, JavaScript, and DSA Question Set - 1 (Answer Key)

```
}  
return true;  
}
```

Question-10

Rotate array elements by k positions.

Answer:

```
function rotateArray(arr, k) {  
  k = k % arr.length;  
  return arr.slice(-k).concat(arr.slice(0, -k));  
}  
  
const arr = [1,2,3,4,5];  
const k = 2;  
console.log(rotateArray(arr, k)); // [4,5,1,2,3]
```