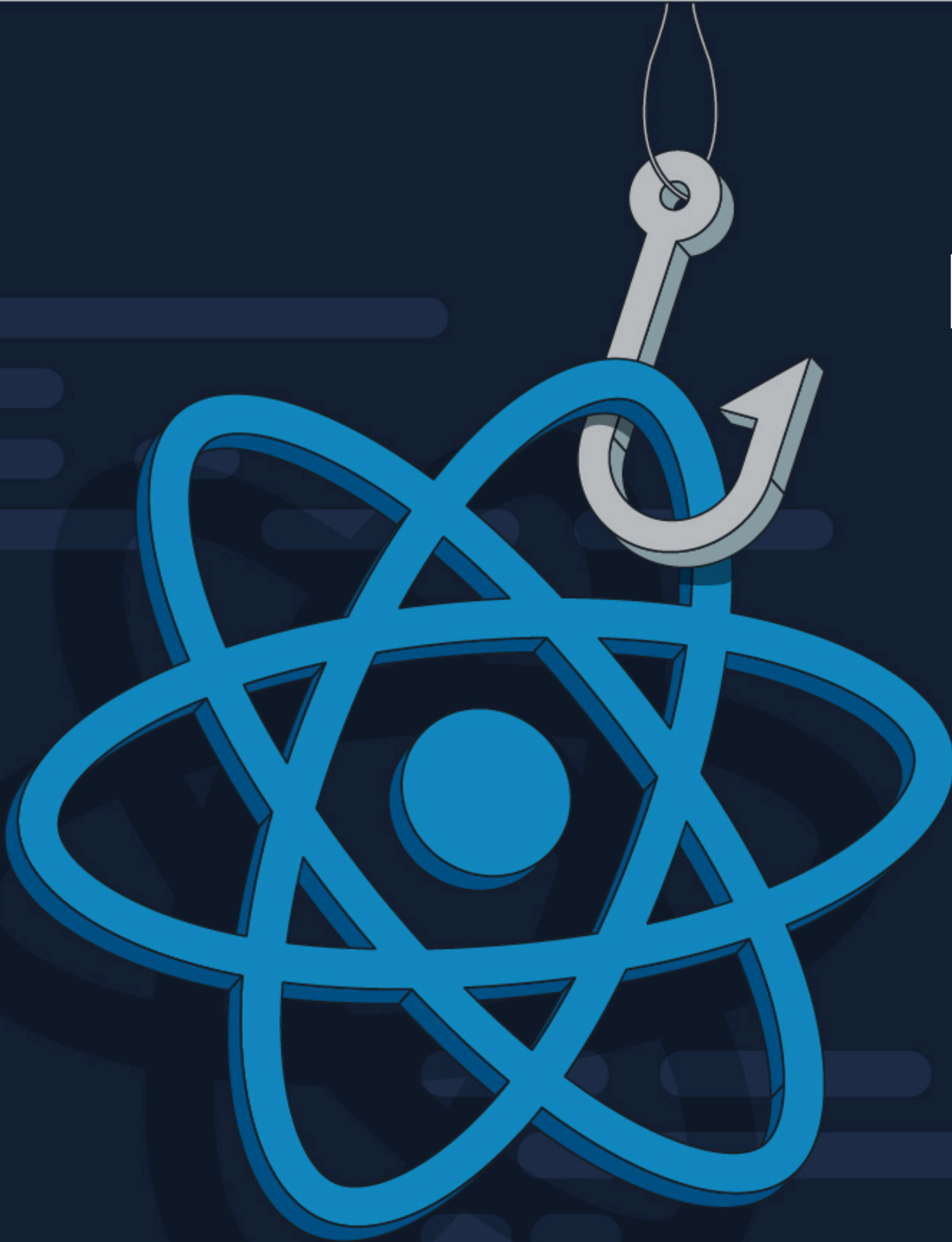


React 19

HOOKS

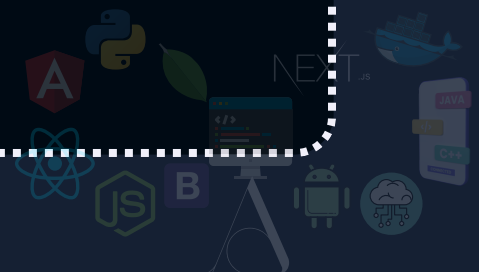


1 useFormStatus

This hook allows you to monitor the status of a form submission (e.g., loading, success, or error).



```
1  import { useFormStatus } from 'react';
2
3  function MyFormComponent() {
4    const { pending, error, success } = useFormStatus();
5
6    return (
7      <form>
8        {pending && <p>Submitting...</p>}
9        {error && <p>Error: {error.message}</p>}
10       {success && <p>Form submitted successfully!</p>}
11       {/* Form fields here */}
12     </form>
13   );
14 }
```



2 useActionState

This hook tracks the state of a specific server action (e.g., loading, success, or error).



```
1  import { useActionState } from 'react';
2
3  function MyActionComponent() {
4    const { pending, error, success } = useActionState('myActionName');
5
6    return (
7      <div>
8        {pending && <p>Action in progress...</p>}
9        {error && <p>Error: {error.message}</p>}
10       {success && <p>Action completed successfully!</p>}
11       {/* Action trigger here */}
12     </div>
13   );
14 }
```

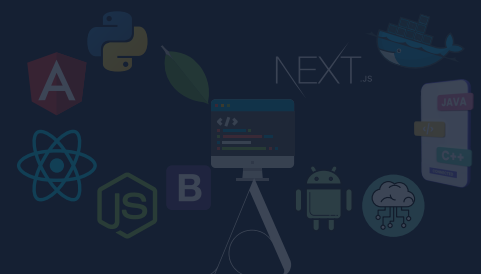


3 useOptimistic

This hook manages optimistic updates by syncing state changes with expected server results.



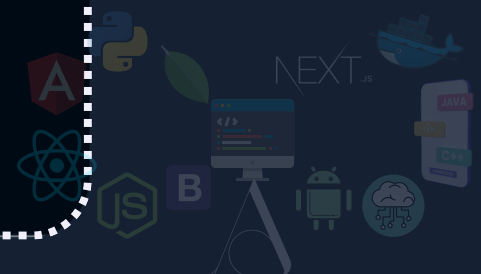
```
1  import { useOptimistic } from 'react';
2
3  function MyOptimisticComponent() {
4    const [optimisticState, setOptimisticState] = useOptimistic(
5      { count: 0 },
6      (currentState, newUpdate) => ({ count: currentState.count + newUpdate })
7    );
8
9    return (
10      <div>
11        <p>Count: {optimisticState.count}</p>
12        <button onClick={() => setOptimisticState(1)}>Increment</button>
13      </div>
14    );
15  }
16
```



4 use

This hook enables Suspense for server data fetching or asynchronous operations.

```
1  import { use } from 'react';
2
3  async function fetchData() {
4    const response = await fetch('/api/data');
5    return response.json();
6  }
7
8  function MyDataComponent() {
9    const data = use(fetchData());
10
11    return (
12      <div>
13        <p>Data: {JSON.stringify(data)}</p>
14      </div>
15    );
16  }
```



Thank you !

Let's connect and make the most of our
community!



SHAIK AMREEN KOUSAR

Comment with your insights.

