Error Boundaries

Error boundaries are React components that catch JavaScript errors anywhere in their child component tree, log those errors, and display a fallback UI instead of the component tree that crashed. Error boundaries catch errors during rendering, in lifecycle methods, and in constructors of the whole tree below them.

A class component becomes an error boundary if it defines either (or both) of the lifecycle methods *static getDerivedStateFromError()* or *componentDidCatch()*.

Error Boundaries

Error boundaries do not catch errors for:

- Event handlers
- Asynchronous code (e.g. setTimeout or requestAnimationFrame callbacks)
- Server side rendering
- Errors thrown in the error boundary itself (rather than its children)

static getDerivedStateFromError()

This lifecycle method is invoked after an error has been thrown by a descendant component. It receives the error that was thrown as a parameter and should return a value to update state.

Use static getDerivedStateFromError() to render a fallback UI after an error has been thrown.

Syntax:-

```
static getDerivedStateFromError(error){

}
```

componentDidCatch()

This lifecycle method is invoked after an error has been thrown by a descendant component.
Use componentDidCatch() to log error information.

Syntax:-

```
componentDidCatch(error, info) {


}
```

Where,
error - The error that was thrown.
info - An object with a componentStack key containing information about which component threw the error.