**4.React-HOL**

**1. Explain the need and benefits of component lifecycle**

The component lifecycle in React represents the phases a component goes through from creation to removal. It is essential for:

* Managing side effects (e.g., API calls)
* Initializing and cleaning up resources
* Improving performance
* Handling errors gracefully
* Performing actions at specific points in a component’s life (mounting, updating, unmounting)

**Benefits:**

* Efficient UI updates
* Cleaner component structure
* Helps manage asynchronous operations
* Better debugging and error handling using lifecycle hooks

**2. Identify various lifecycle hook methods**

React class components support the following **lifecycle hook methods**:

| **Phase** | **Lifecycle Methods** |
| --- | --- |
| Mounting | constructor(), static getDerivedStateFromProps(), render(), componentDidMount() |
| Updating | shouldComponentUpdate(), render(), getSnapshotBeforeUpdate(), componentDidUpdate() |
| Unmounting | componentWillUnmount() |
| Error Handling | componentDidCatch() |

**3. List the sequence of steps in rendering a component**

**Component Lifecycle Order (Mounting phase):**

1. constructor()
2. static getDerivedStateFromProps()
3. render()
4. componentDidMount()

**On update:**

1. static getDerivedStateFromProps()
2. shouldComponentUpdate()
3. render()
4. getSnapshotBeforeUpdate()
5. componentDidUpdate()

**On error:**

1. componentDidCatch()

**HANDS-ON**

**CODE: Post.js**

class Post {

  constructor(id, title, body) {

    this.id = id;

    this.title = title;

    this.body = body;

  }

}

export default Post;

**Posts.js**

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

      hasError: false,

    };

  }

  loadPosts = () => {

    fetch('https://jsonplaceholder.typicode.com/posts')

      .then(response => response.json())

      .then(data => {

        const postList = data.slice(0, 10).map(post =>

          new Post(post.id, post.title, post.body)

        );

        this.setState({ posts: postList });

      })

      .catch(error => {

        console.error('Error fetching posts:', error);

        this.setState({ hasError: true });

      });

  };

  componentDidMount() {

    this.loadPosts();

  }

  componentDidCatch(error, info) {

    alert('Something went wrong!');

    console.log(error, info);

  }

  render() {

    return (

      <div>

        <h1>Blog Posts</h1>

        {this.state.posts.map(post => (

          <div key={post.id}>

            <h3>{post.title}</h3>

            <p>{post.body}</p>

          </div>

        ))}

      </div>

    );

  }

}

export default Posts;

**App.js**

import React from 'react';

import Posts from './Posts';

function App() {

  return (

    <div className="App">

      <Posts />

    </div>

  );

}

export default App;

**OUTPUT**

