

## 4.React.js – HOL

### 1. Explain the Need and Benefits of Component Lifecycle

React components go through a lifecycle from creation, updating, and unmounting. Lifecycle methods allow developers to:

- Perform operations after component mounts (e.g., API calls)
  - Handle errors gracefully
  - Optimize performance
  - Control component behavior during different phases
- 

### 2. Identify Various Lifecycle Hook Methods

Common Class Component Lifecycle Methods:

- **Mounting Phase**
    - constructor()
    - componentDidMount()
  - **Updating Phase**
    - componentDidUpdate()
  - **Unmounting Phase**
    - componentWillUnmount()
  - **Error Handling**
    - componentDidCatch(error, info)
- 

### 3. List Sequence of Steps in Rendering a Component

1. constructor()

2. render()
3. componentDidMount()
4. componentDidUpdate() (when props/state changes)
5. componentWillUnmount() (before removal)
6. componentDidCatch() (on error)

**Command** : npx create-react-app blogapp

### **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 postObjects = data.slice(0, 10).map(item => new Post(item.id, item.title, item.body));

      this.setState({ posts: postObjects });

    })

    .catch(error => {

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

    });

}

componentDidMount() {

  this.loadPosts();

}

componentDidCatch(error, info) {

  console.error("Error caught in Posts component:", error, info);

  alert("An error occurred in the Posts component.");

  this.setState({ hasError: true });

}

render() {

  if (this.state.hasError) {
```

```

    return <h2>Something went wrong while displaying posts.</h2>;
  }

  return (
    <div>

      <h1>Blog Posts</h1>

      {this.state.posts.map(post => (
        <div key={post.id} style={{ border: "1px solid #ccc", padding: "10px", margin: "10px 0" }}>

          <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;
```

**Command :** npm start

**Output :**



## Blog Posts

### sunt aut facere repellat provident occaecati excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

### qui est esse

est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

### ea molestias quasi exercitationem repellat qui ipsa sit aut

et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

### eum et est occaecati

ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum illo velit

### nesciunt quas odio

repudiandae veniam quaerat sunt sed alias aut fugiat sit autem sed est voluptatem omnis possimus esse voluptatibus quis est aut tenetur dolor neque