**WEEK 6**

**2. ReactJS-HOL**

**Component Lifecycle Methods**

**Step 1: Create a New React Application**

npx create-react-app blogapp

**Step 2:** Navigate to **blogapp** folder and open it.

**Step 3: Create Post.js File**

import React from 'react';

function Post({ title, body }) {

return (

<div className="post">

<h2>{title}</h2>

<p>{body}</p>

</div>

);

}

export default Post;

**Step 4: Create Posts Component (Class-Based)**

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

};

}

// Method to load posts

loadPosts() {

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

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

.then(data => this.setState({ posts: data }))

.catch(error => this.setState({ error }));

}

// Lifecycle method: componentDidMount

componentDidMount() {

this.loadPosts();

}

// Lifecycle method: componentDidCatch

componentDidCatch(error, info) {

alert('An error occurred: ' + error);

console.error('Error details:', info);

}

render() {

const { posts, error } = this.state;

if (error) {

return <h3>Error loading posts!</h3>;

}

return (

<div>

<h1>Blog Posts</h1>

{posts.map(post => (

<Post key={post.id} title={post.title} body={post.body} />

))}

</div>

);

}

}

export default Posts;

**Step 5: Add Posts Component in App.js**

import React from 'react';

import Posts from './Posts';

function App() {

return (

<div>

<Posts />

</div>

);

}

export default App;