## React hand-on 4

Create a new react application using create-react-app tool with the name as "blogapp" Open the application using VS Code Create a new file named as Post.js in src folder with following properties

```
Step1: create a react app
Step2: create post.js file in src
class Post {
constructor(id, title, body) {
 this.id = id;
 this.title = title;
 this.body = body;
}
}
export default Post;
Step3: create posts.js file in src
import React, { Component } from 'react';
import Post from './Post';
class Posts extends Component {
 constructor(props) {
 super(props);
```

```
this.state = {
  posts: [],
   hasError: false
 };
}
// Fetch posts from API
 loadPosts = async () => {
 try {
  const response = await
fetch("https://jsonplaceholder.typicode.com/posts");
   const data = await response.json();
  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 });
 }
};
// Lifecycle hook – gets called after initial render
 componentDidMount() {
```

```
this.loadPosts();
}
// Error boundary – catches rendering errors in children
componentDidCatch(error, info) {
 alert("An error occurred while rendering the posts.");
 console.error("Caught by componentDidCatch:", error, info);
 this.setState({ hasError: true });
}
render() {
 if (this.state.hasError) {
 return <h2>Something went wrong!</h2>;
 }
 return (
  <div style={{ padding: '20px' }}>
   <h1>Blog Posts</h1>
   {this.state.posts.map(post => (
    <div key={post.id} style={{ marginBottom: '20px' }}>
     <h3>{post.title}</h3>
     {post.body}
    </div>
```

```
))}
  </div>
 );
}
}
export default Posts;
//the link that gives a latin-like placehilder text which is not a real language
// it will fetch the dummy data
Step4: in App.js
import React from 'react';
import './App.css';
import Posts from './Posts';
function App() {
 return (
 <div className="App">
  <Posts/>
  </div>
);
}
export default App;
```

# Step5: run the app using. "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

#### aui 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

## dolorem eum magni eos aperiam quia

ut aspernatur corporis harum nihil quis provident sequi mollitia nobis aliquid molestiae perspiciatis et ea nemo ab reprehenderit accusantium quas voluptate dolores velit et doloremque molestiae

## magnam facilis autem

dolore placeat quibusdam ea quo vitae magni quis enim qui quis quo nemo aut saepe quidem repellat excepturi ut quia sunt ut sequi eos ea sed quas

## dolorem dolore est ipsam

dignissimos aperiam dolorem qui eum facilis quibusdam animi sint suscipit qui sint possimus cum quaerat magni maiores excepturi ipsam ut commodi dolor voluptatum modi aut vitae

## nesciunt iure omnis dolorem tempora et accusantium

consectetur animi nesciunt iure dolore enim quia ad veniam autem ut quam aut nobis et est aut quod aut provident voluptas autem voluptas