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

Figure 2: Post class

1. Create a new class based component named as **Posts** inside **Posts.js** file

Figure 3: Posts Component

1. Initialize the component with a list of Post in state of the component using the constructor
2. Create a new method in component with the name as **loadPosts()** which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url (<https://jsonplaceholder.typicode.com/posts>)

Figure 4: loadPosts() method

1. Implement the **componentDidMount()** hook to make calls to **loadPosts()** which will fetch the posts

Figure 5: componentDidMount() hook

1. Implement the **render()** which will display the title and post of posts in html page using heading and paragraphs respectively.

Figure 6: render() method

1. Define a **componentDidCatch()** method which will be responsible for displaying any error happing in the component as alert messages.

Figure 7: componentDidCatch() hook

1. Add the Posts component to App component.
2. Build and Run the application using *npm start* command.

**CODES:**

**Src/Post.js**

import React, { Component } from 'react';

class Post extends Component {

render() {

const { title, body } = this.props;

return (

<div className="post">

<h3>{title}</h3>

<p>{body}</p>

</div>

);

}

}

export default Post;

**src/Posts.js**

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

constructor(props) {

super(props);

this.state = {

posts: [],

hasError: false,

errorMessage: ''

};

}

loadPosts = () => {

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

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

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

.catch(error => {

this.setState({ hasError: true, errorMessage: error.message });

});

};

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

this.setState({ hasError: true, errorMessage: error.toString() });

alert(`Error occurred: ${error.toString()}`);

}

render() {

const { posts, hasError, errorMessage } = this.state;

if (hasError) {

return <h2>Error: {errorMessage}</h2>;

}

return (

<div>

<h1>Blog Posts</h1>

{posts.slice(0, 10).map(post => (

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

))}

</div>

);

}

}

export default Posts;

**src/app.js**

import React from 'react';

import Posts from './Posts';

function App() {

return (

<div className="App">

<Posts />

</div>

);

}

export default App;

**OUTPUTS:**

****

****