**4. ReactJS HandsOn**

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

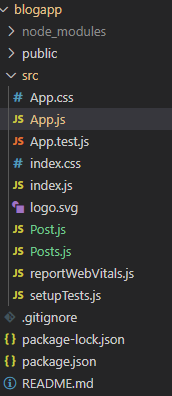
1. Define a **componentDidCatch()** method which will be responsible for displaying any error happing in the component as alert messages.



Figure 7: componentDidCatch() hook

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

**Project Structure:**



**Post.js**

// src/Post.js

class Post {

  constructor(id, title, body) {

    this.id = id;

    this.title = title;

    this.body = body;

  }

}

export default Post;

**Posts.js**

import React from "react";

import Post from "./Post";

class Posts extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [

        new Post(1, "Initial Post", "This is a placeholder post."),

        new Post(2, "Another Post", "Here’s some sample content."),

      ],

    };

  }

  loadPosts() {

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

      .then((res) => res.json())

      .then((data) => {

        const postObjects = data.slice(0, 5).map(

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

        );

        this.setState({ posts: postObjects });

      })

      .catch((error) => {

        console.error("Failed to fetch posts", error);

      });

  }

  componentDidMount() {

    this.loadPosts();

  }

  componentDidCatch(error, info) {

    alert("An error occurred while rendering posts.");

    console.error("Error:", error, info);

  }

  render() {

    return (

      <div style={{ padding: "1rem" }}>

        <h2>Posts</h2>

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

          <div

            key={post.id}

            style={{

              border: "1px solid #ccc",

              marginBottom: "1rem",

              padding: "1rem",

              borderRadius: "5px",

              backgroundColor: "#f9f9f9",

            }}

          >

            <h3>{post.title}</h3>

            <p>{post.body}</p>

          </div>

        ))}

      </div>

    );

  }

}

export default Posts;

**Output:**

