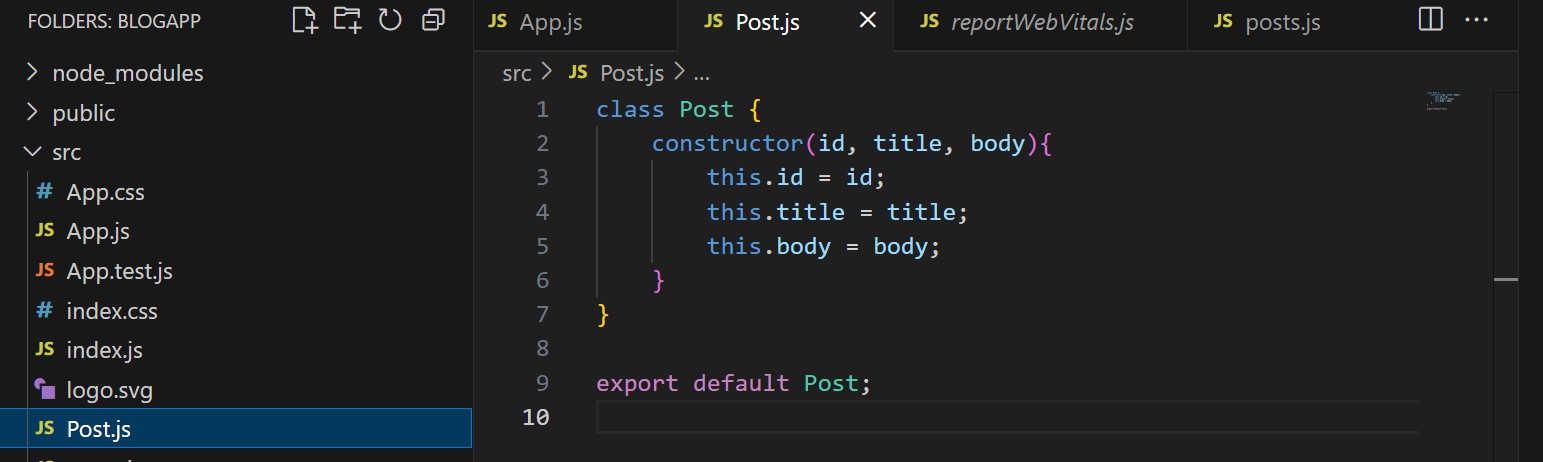
4. ReactJS-HOL

1. Create a new react application using create-react-app tool with the name as “blogapp”

PS C:\Users\lenovo> npx create-react-app blogapp

1. Open the application using VS Code
2. Create a new file named as Post.js in src folder with following properties



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

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>)
3. Implement the componentDidMount() hook to make calls to loadPosts() which will fetch the posts
4. Implement the render() which will display the title and post of posts in html page using heading and paragraphs respectively.
5. Define a componentDidCatch() method which will be responsible for displaying any error happing in the component as alert messages.

import React from 'react'; import Post from './Post';

class Posts extends React.Component { constructor(props) {

super(props); this.state = { posts: []

};

}

// Correctly defined class method loadPosts = () => {

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

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

.then(data => {

this.setState({ posts: data });

})

.catch(error => {

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

});

};

// Lifecycle method componentDidMount() {

this.loadPosts();

}

// Error boundary componentDidCatch(error, info) {

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

}

render() { return (

<div>

<h2>Posts</h2>

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

<div

key={post.id}

style={{ border: "1px solid #ccc", margin: "10px", padding: "10px" }}

>

<h3>{post.title}</h3>

<p>{post.body}</p>

</div>

))}

</div>

);

}

}

export default Posts;

1. Add the Posts component to App component

import React from 'react'; import Posts from './posts';

function App() { return (

<div className="App">

<Posts />

</div>

);

}

export default App;

# Build and Run the application using npm start command.

PS C:\Users\lenovo\blogapp> npm start

* + [blogapp@0.1.0](mailto:blogapp@0.1.0) start
  + react-scripts start

Compiled successfully!

You can now view blogapp in the browser.

Compiled with warnings.

[eslint]

src\posts.js

Line 2:8: 'Post' is defined but never used no-unused-vars

Search for the keywords to learn more about each warning.

To ignore, add // eslint-disable-next-line to the line before.

WARNING in [eslint]

src\posts.js

Line 2:8: 'Post' is defined but never used no-unused-vars

webpack compiled with 1 warning

Output:

