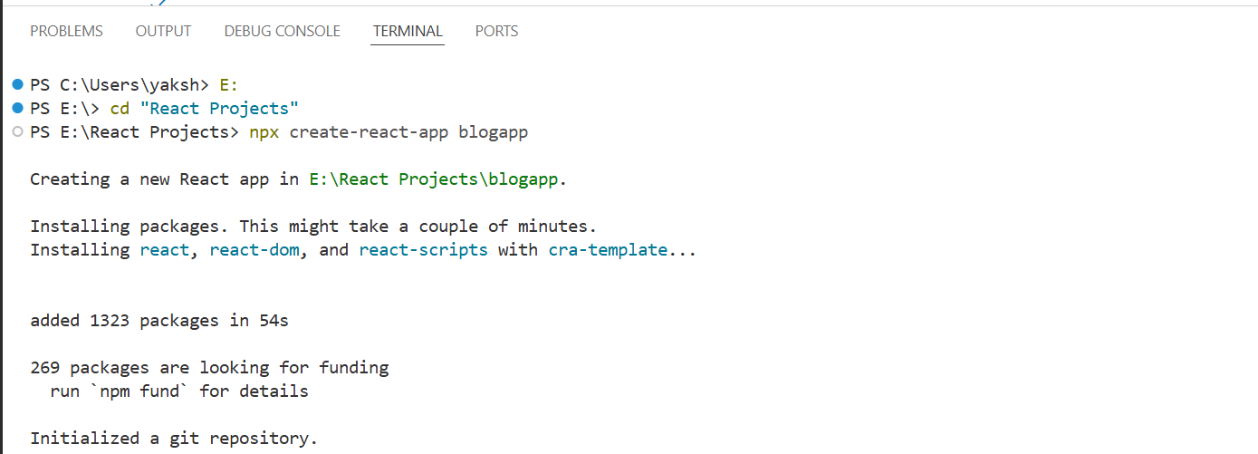
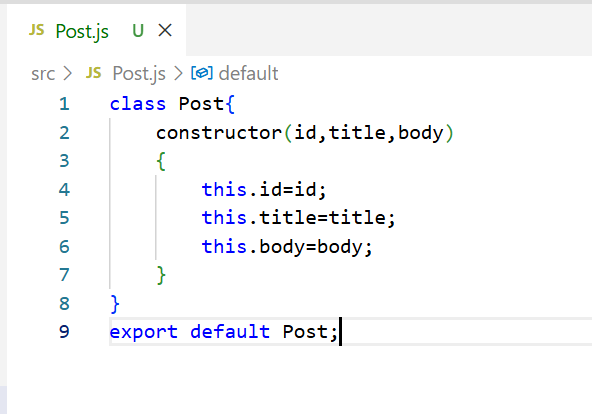
**BlogApp using React**

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



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



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

import React from "react";

import Post from "./Post";

class Posts extends React.Component {

constructor(props) {

super(props);

this.state = {

posts: [],

};

}

loadPosts = async () => {

try {

const response = await fetch("https://jsonplaceholder.typicode.com/posts");

const data = await response.json();

const postObjects = data.map((p) => new Post(p.id, p.title, p.body));

this.setState({ posts: postObjects });

} catch (error) {

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

}

};

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

alert("An error occurred in the Posts component: " + error);

console.error(error, info);

}

render() {

return (

<div>

<h2>Blog Posts</h2>

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

<div key={post.id} style={{ marginBottom: "20px" }}>

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

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

</div>

))}

</div>

);

}

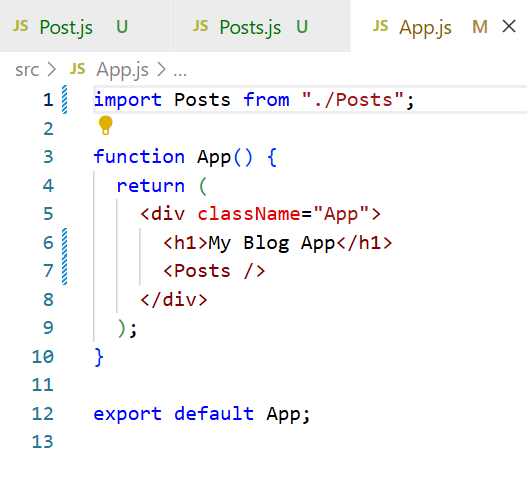
}

export default Posts;

Contents of Posts Component :

* An empty array of Post objects for state management.
* loadPosts() method that fetches data from https://jsonplaceholder.typicode.com/posts, converts each item into a Post object, and updates the component state.
* componentDidMount() lifecycle method to call loadPosts() automatically when the component is mounted.
* render() to display each post’s title and body using heading and paragraph tags.
* componentDidCatch() to handle and alert any runtime errors within the component.

1. Added the Posts component to App component.



1. After running the application using npm start. Open browser and type localhost:3000 to see the result.

