

To integrate the previous response with the given HTML code, you would need to do the following:

1. Remove the existing `form` element and the `onclick` event listener for the "Read Posts" button.
2. Add a `div` element with the ID of `root` to the HTML code. This is where the React app will be rendered.
3. Replace the existing script tags with the following code:

php

 Copy code

```
<script src="https://unpkg.com/react/umd/react.development.js"></script> <script
src="https://unpkg.com/react-dom/umd/react-dom.development.js"></script> <script
src="https://unpkg.com/axios/dist/axios.min.js"></script> <div id="root"></div> <script
type="module"> import React from 'react'; import ReactDOM from 'react-dom'; import App
from './assn4-app/src/App'; ReactDOM.render(<App />, document.getElementById('root'));
</script> <script type="text/javascript"> const form = document.querySelector("form");
form.addEventListener("submit", e => { e.preventDefault(); const formData = new
FormData(form); const topic = formData.get("topic"); const data = formData.get("data");
fetch("/addPost", { method: "POST", headers: { "Content-Type": "application/json" },
body: JSON.stringify({ topic, data }) }) .then(response => response.json()) .then(result
=> console.log(result)) .catch(error => console.error(error)); }); </script>
```

4. Modify the `App` component as follows:

javascript

 Copy code

```
import React, { useState, useEffect } from 'react'; import axios from 'axios'; const App
= () => { const [posts, setPosts] = useState([]); const [topic, setTopic] = useState('');
const [data, setData] = useState(''); useEffect(() => { axios.get('/getPosts') .then(res
=> setPosts(res.data)) .catch(err => console.log(err)); }, []); const handleSubmit = (e)
=> { e.preventDefault(); axios.post('/addPost', { topic, data }) .then(() => {
setTopic(''); setData(''); }) .catch(err => console.log(err)); }; return ( <div>
<h1>Topic & Data</h1> <form onSubmit={handleSubmit}> <label htmlFor="topic">Topic:
</label><br /> <input type="text" id="topic" name="topic" value={topic} onChange={e =>
setTopic(e.target.value)} /><br /> <label htmlFor="data">Data:</label><br /> <input
type="text" id="data" name="data" value={data} onChange={e => setData(e.target.value)} />
<br /> <button type="submit">Submit</button> </form> <div> {posts.map(post => ( <div key=
{post._id}> <h2>{post.topic}</h2> <p>{post.data}</p> </div> ))} </div> </div> ); };
export default App;
```

This code uses the `useState` and `useEffect` hooks to manage the `posts`, `topic`, and `data` states. It fetches the initial set of posts from the server using the `useEffect` hook and the `axios` library. It also

defines a **handleSubmit** function that is called when the form is submitted. This function sends a POST request to the server using the **axios** library and updates the **topic** and **data** states.