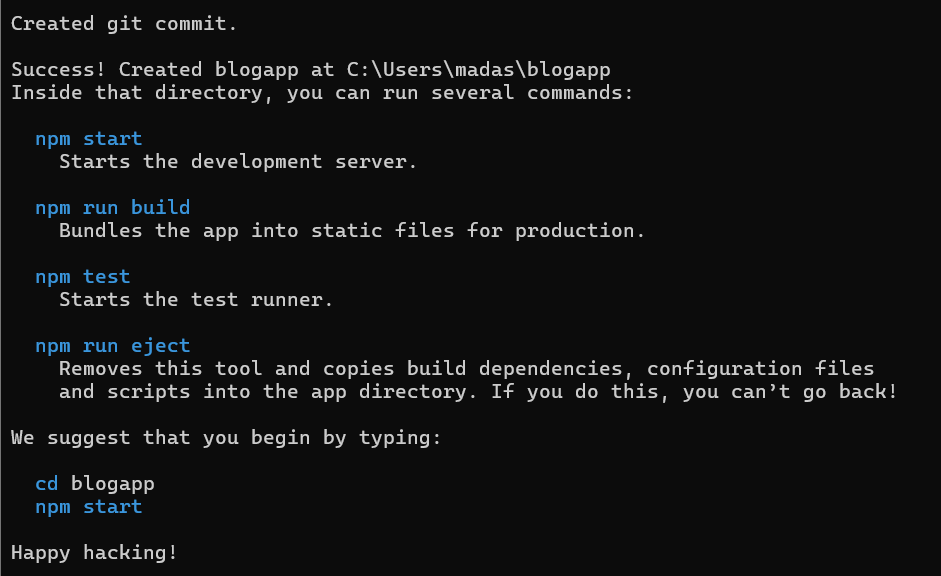
**Superset ID: 6394725**

**ReactJS-HOL**

**HANDS ON-4:**

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



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

class Post{

constructor(id,title,body){

this.id=id;

this.title=title;

this.body=body;

}

}

export default Post;

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

class Posts extends React.Component {

constructor(props) {

super(props); };

}

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>)

class Posts extends React.Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

};

}

loadPosts() {

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

.then((response) => {

if (!response.ok) {

throw new Error('Failed to fetch posts');

}

return response.json();

})

.then((data) => {

const postObjects = data.map(

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

);

this.setState({ posts: postObjects });

})

.catch((error) => {

this.setState({ error });

});

}

}

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

class Posts extends React.Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

};

}

loadPosts() {

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

.then((response) => {

if (!response.ok) {

throw new Error('Failed to fetch posts');

}

return response.json();

})

.then((data) => {

const postObjects = data.map(

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

);

this.setState({ posts: postObjects });

})

.catch((error) => {

this.setState({ error });

});

}

componentDidMount() {

this.loadPosts();

}

}

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

class Posts extends React.Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

};

}

loadPosts() {

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

.then((response) => {

if (!response.ok) {

throw new Error('Failed to fetch posts');

}

return response.json();

})

.then((data) => {

const postObjects = data.map(

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

);

this.setState({ posts: postObjects });

})

.catch((error) => {

this.setState({ error });

});

}

componentDidMount() {

this.loadPosts();

}

render() {

const { posts } = this.state;

return (

<div>

<h1>Blog Posts</h1>

{posts.map((post) => (

<div key={post.id}>

<h2>{post.title}</h2>

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

<hr />

</div>

))}

</div>

);

}

}

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

class Posts extends React.Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

};

}

loadPosts() {

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

.then((response) => {

if (!response.ok) {

throw new Error('Failed to fetch posts');

}

return response.json();

})

.then((data) => {

const postObjects = data.map(

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

);

this.setState({ posts: postObjects });

})

.catch((error) => {

this.setState({ error });

});

}

componentDidMount() {

this.loadPosts();

}

render() {

const { posts } = this.state;

return (

<div>

<h1>Blog Posts</h1>

{posts.map((post) => (

<div key={post.id}>

<h2>{post.title}</h2>

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

<hr />

</div>

))}

</div>

);

}

componentDidCatch(error, info) {

alert('An error occurred: ' + error.message);

}

}

1. Add the Posts component to App component.

import logo from './logo.svg';

import './App.css';

import React from 'react';

import Posts from './Posts';

function App() {

return (

<div className="App">

<Posts />

</div>

);

}

export default App;

1. Build and Run the application using *npm start* command.

