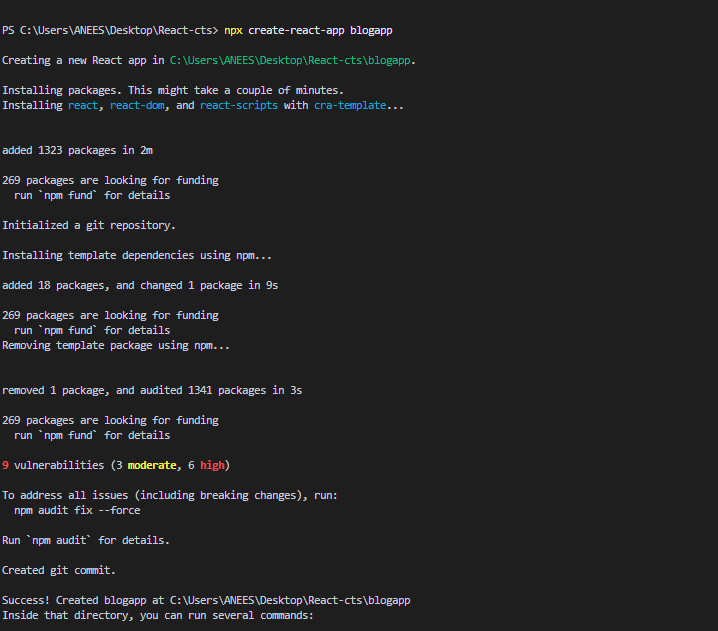
**4. ReactJS-HOL**

In this hands-on lab, you will learn how to:

* Implement componentDidMount() hook
* Implementing componentDidCatch() life cycle hook.

**Steps:**

* Create a New React App named “blogapp”



* Create Post.js inside src folder

**Post.js**

class Post {

constructor(id, title, body) {

this.id = id;

this.title = title;

this.body = body;

}

}

export default Post;

* Create Posts.js inside src folder

**Posts.js**

import React from 'react';

import Post from './Post';

class Posts extends React.Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

}; }

loadPosts() {

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

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

.then(data => {

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

this.setState({ posts: postList });

})

.catch(error => {

this.setState({ error });

});

}

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

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

}

render() {

return (

<div>

<h1>Blog Posts</h1>

{this.state.error && <p style={{ color: 'red' }}>{this.state.error.message}</p>}

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

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

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

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

<hr />

</div>

))}

</div> ); }}

export default Posts;

* Edit App.js

App.js

import React from 'react';

import Posts from './Posts';

function App() {

return (

<div className="App">

<Posts />

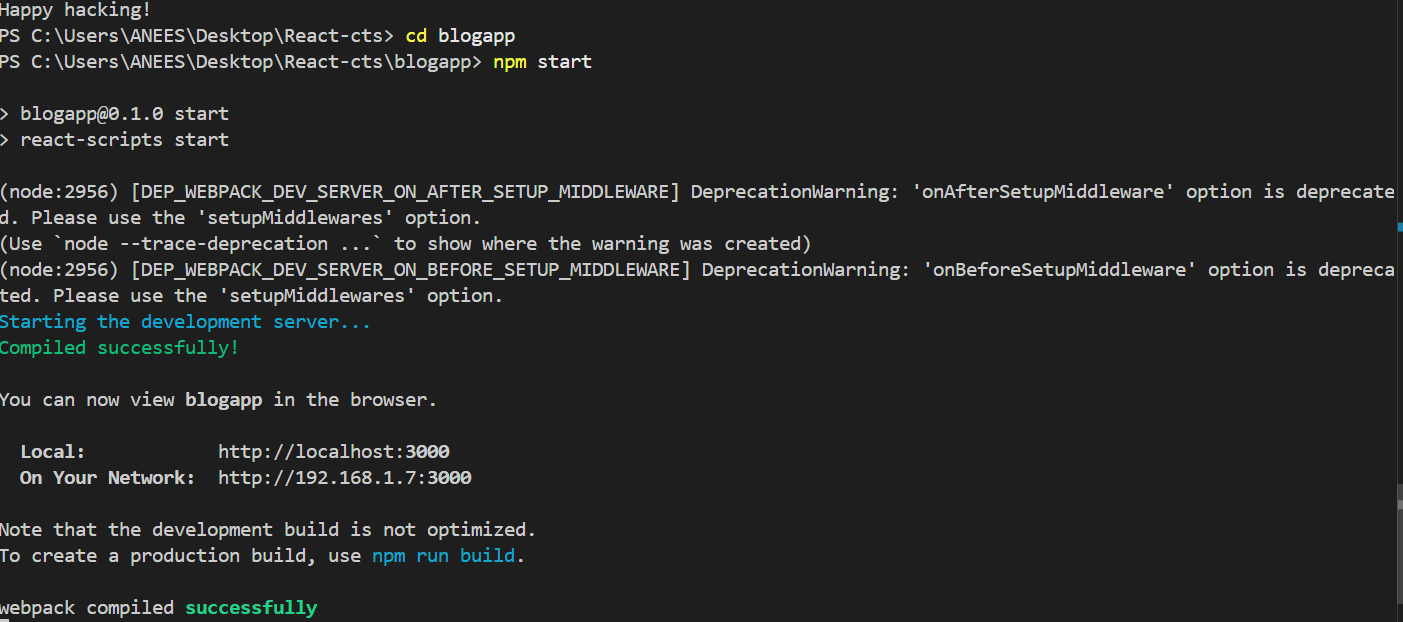
</div>

);

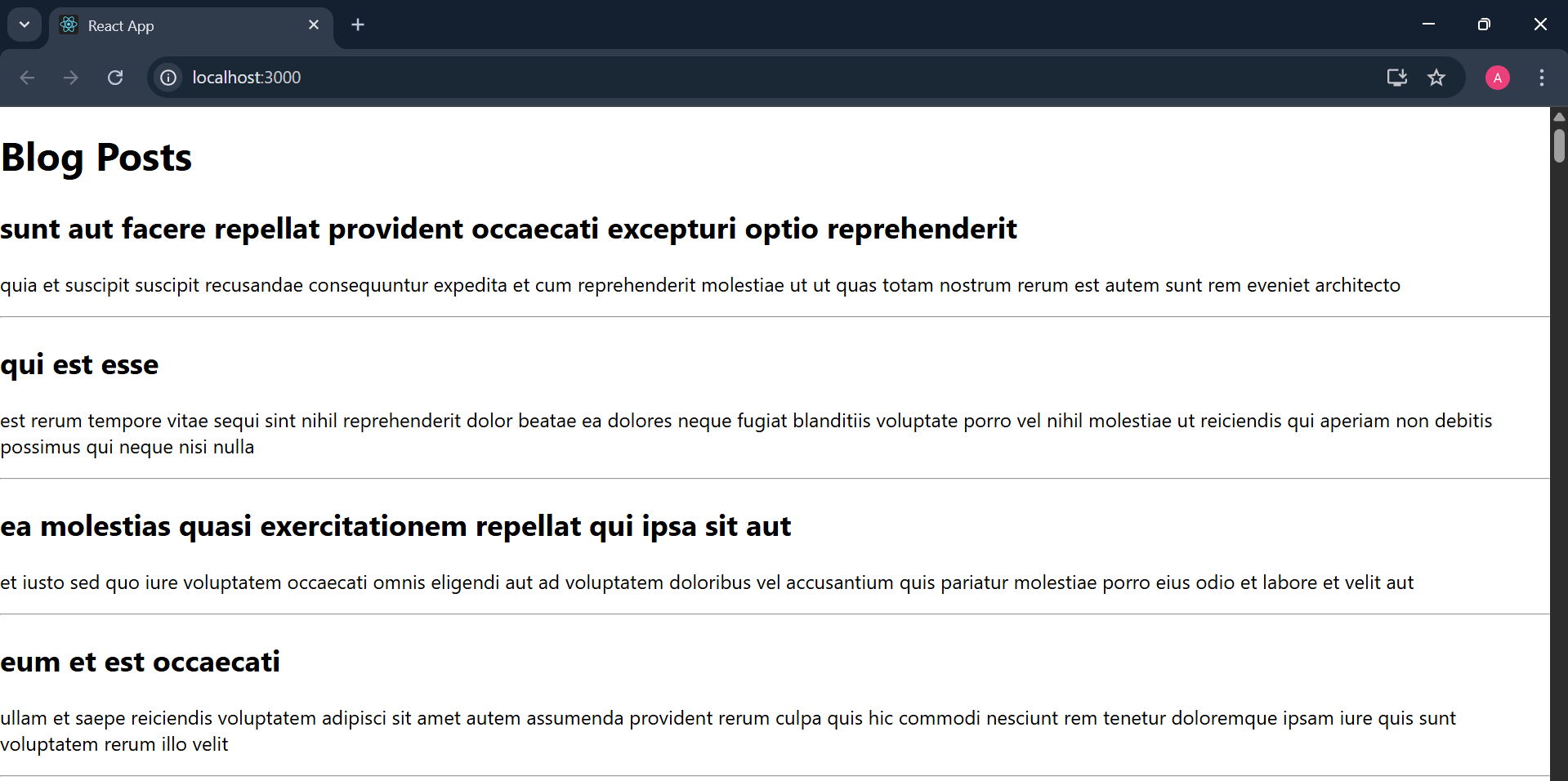
}

export default App;

* Run the application  
  Command: npm start



**Output**

****