Name: Amogh Girish Nagarkar

Super Set ID: 6403503

**Objectives**

* Explain the need and benefits of the **component lifecycle**.
* Identify various **lifecycle hook methods**.
* List the **sequence of steps** in rendering a component.

**Steps**

**1. Create a React application**

npx create-react-app blogapp

**2. Open the application in VS Code**

cd blogapp

code .

**3. Create Post.js**

import React from 'react';

function Post({ title, body }) {

return (

<div className="post">

<h2>{title}</h2>

<p>{body}</p>

</div>

);

}

export default Post;

**4. Create Posts.js**

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

constructor(props) {

super(props);

this.state = {

posts: [],

hasError: false

};

}

loadPosts() {

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

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

.then(data => this.setState({ posts: data }))

.catch(error => {

console.error("Fetch error:", error);

this.setState({ hasError: true });

});

}

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

alert("Error occurred: " + error);

this.setState({ hasError: true });

console.error("Error:", error, info);

}

render() {

if (this.state.hasError) {

return <h1>Something went wrong while loading posts.</h1>;

}

return (

<div>

<h1>Blog Posts</h1>

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

<Post key={post.id} title={post.title} body={post.body} />

))}

</div>

);

}

}

export default Posts;

**5. Edit App.js**

import './App.css';

import Posts from './Posts';

function App() {

return (

<div className="App">

<Posts />

</div>

);

}

export default App;

**6. Optional: Add Basic Styling**

.post {

border: 1px solid #ccc;

padding: 15px;

margin: 10px auto;

width: 80%;

border-radius: 5px;

background-color: #f9f9f9;

}

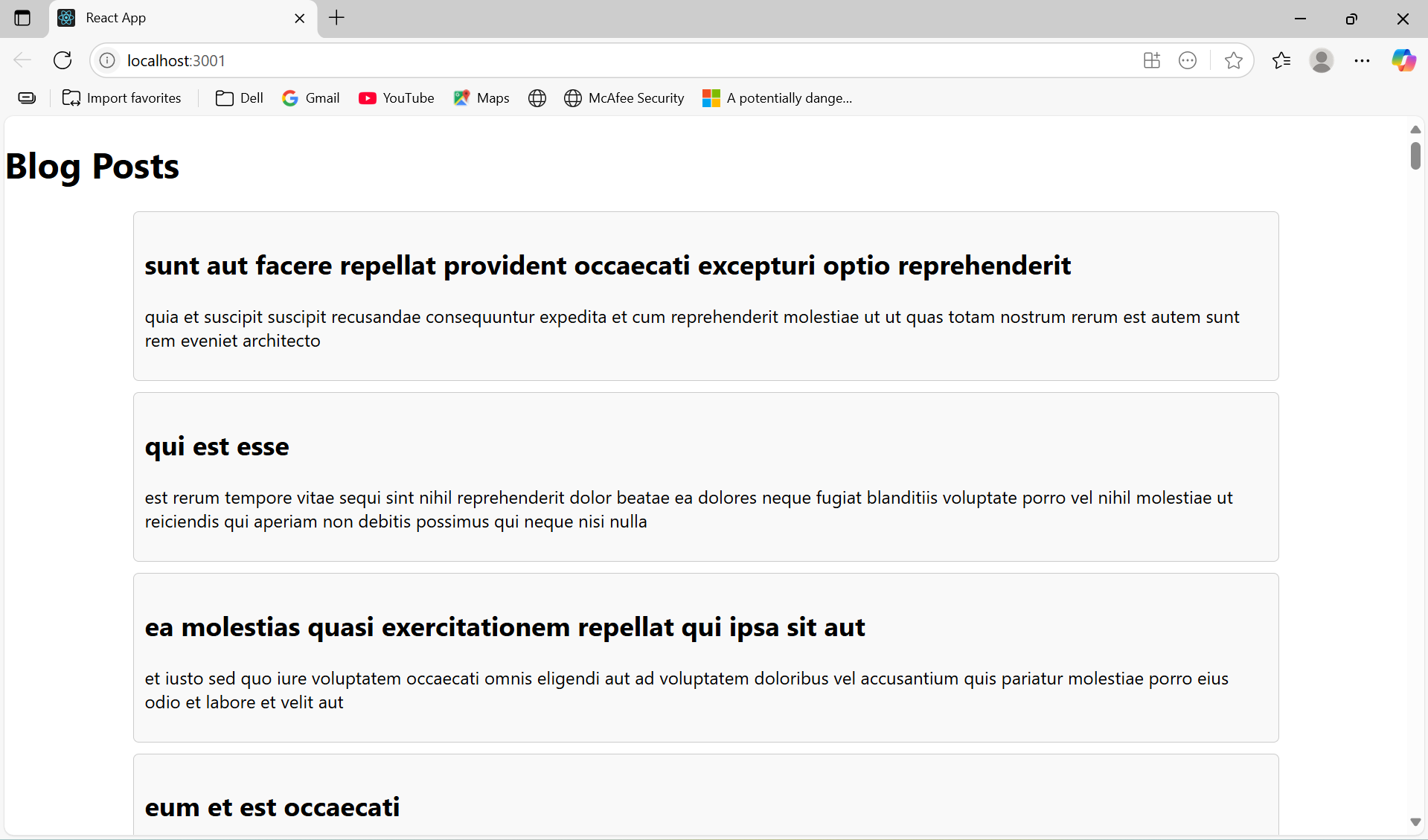
**7. Run the Application**

npm start

**8. Open in Browser**

http://localhost:3001

**Output:**

****