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

**Program**

**Post.js**

// Post.js

class Post {

constructor(userId, id, title, body) {

this.userId = userId;

this.id = id;

this.title = title;

this.body = body;

}

}

export default Post;

**Posts.js**

// Posts.js

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

};

}

// ✅ 6. Method to fetch posts

loadPosts() {

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

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

.then((data) => {

const postList = data.map(

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

);

this.setState({ posts: postList });

})

.catch((err) => {

this.setState({ error: err });

});

}

// ✅ 7. Call loadPosts when component mounts

componentDidMount() {

this.loadPosts();

}

// ✅ 9. Catch and handle errors in child components

componentDidCatch(error, info) {

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

}

// ✅ 8. Render post titles and bodies

render() {

const { posts } = this.state;

return (

<div>

<h1>Blog Posts</h1>

{posts.map((post) => (

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

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

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

</div>

))}

</div>

);

}

}

export default Posts;

**App.js**

import React from 'react';

import './App.css';

import Posts from './Posts';

function App() {

return (

<div className="App">

<Posts />

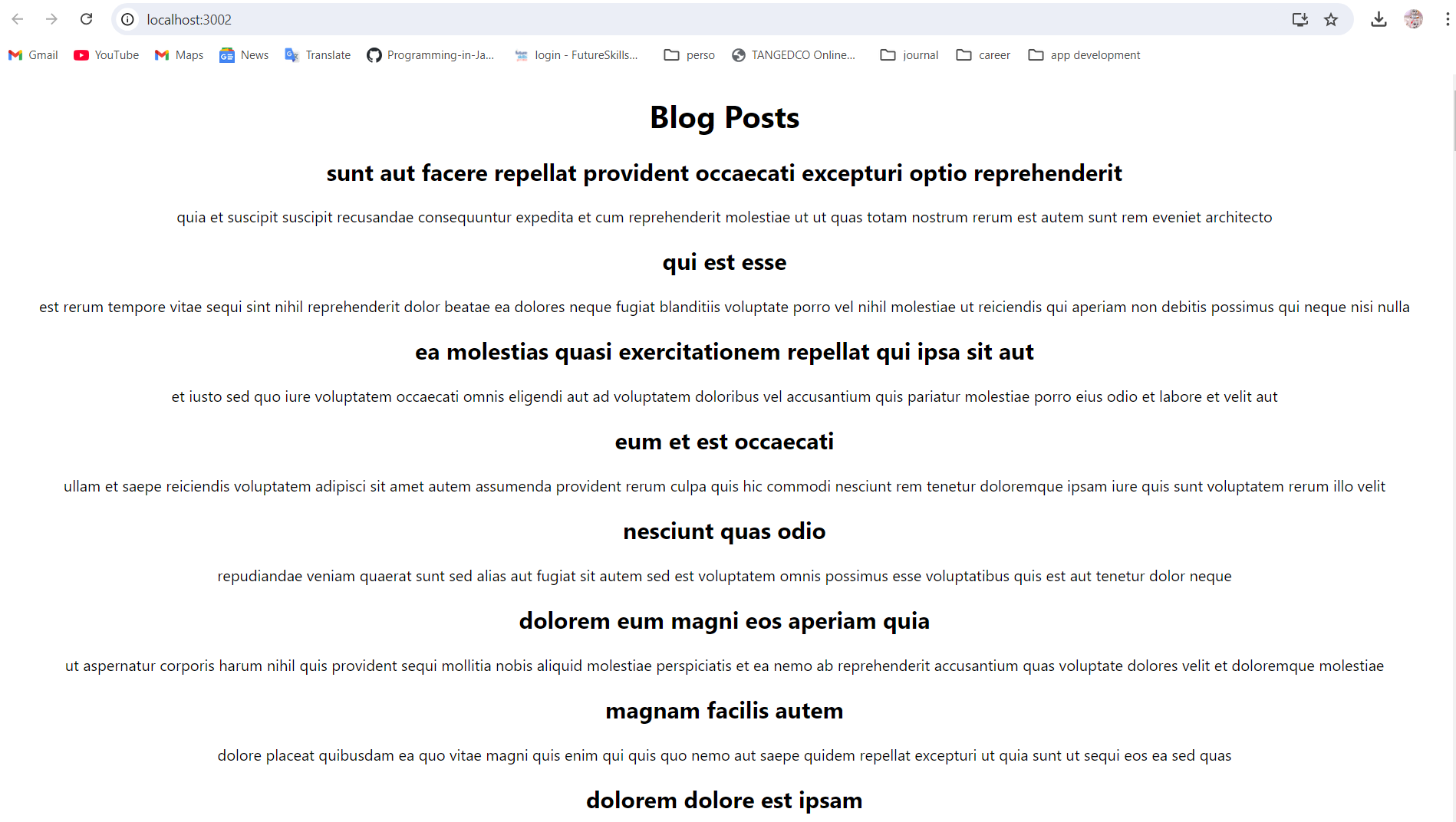
</div>

);

}

export default App;

**Output**

****