

TASK-25

25. Pass State as Props to Child Components:

- Pass the blog data to a `BlogList.jsx` component to display all posts.

1. App Component: **App.js**

```
// src/App.js
import React, { useState } from 'react';
import BlogList from './components/BlogList'; // Import the BlogList component
import './App.css';

function App() {
  // Initial state for blog posts
  const [blogs, setBlogs] = useState([
    { title: 'The Beauty of React', content: 'React is a powerful library...', author: 'Het Dabhi' },
    { title: 'PropTypes in Action', content: 'Using PropTypes ensures better code...', author: 'Dhruv Dabhi' },
    { title: 'Components Simplified', content: 'Components are the building blocks of React.', author: 'Raj Dabhi' },
  ]);

  // New blog form state
  const [newBlog, setNewBlog] = useState({
    title: '',
    content: '',
    author: ''
  });

  // Handle input changes
  const handleInputChange = (e) => {
    const { name, value } = e.target;
    setNewBlog({
      ...newBlog,
      [name]: value,
    });
  };

  // Handle form submission
  const handleSubmit = (e) => {
    e.preventDefault();
    setBlogs([...blogs, newBlog]);
    setNewBlog({ title: '', content: '', author: '' }); // Reset form
  };
}
```

```
return (  
  <div className="App">  
    <h1>Bloggng Platform</h1>  
  
    { /* Blog post form */}  
    <form onSubmit={handleSubmit}>  
      <div>  
        <label htmlFor="title">Title</label>  
        <input  
          type="text"  
          id="title"  
          name="title"  
          value={newBlog.title}  
          onChange={handleInputChange}  
          placeholder="Blog title"  
          required  
        />  
      </div>  
      <div>  
        <label htmlFor="content">Content</label>  
        <textarea  
          id="content"  
          name="content"  
          value={newBlog.content}  
          onChange={handleInputChange}  
          placeholder="Blog content"  
          required  
        />  
      </div>  
      <div>  
        <label htmlFor="author">Author</label>  
        <input  
          type="text"  
          id="author"  
          name="author"  
          value={newBlog.author}  
          onChange={handleInputChange}  
          placeholder="Author name"  
          required  
        />  
      </div>  
      <button type="submit">Add Blog</button>  
    </form>  
  
    { /* Pass blogs state as props to BlogList */}  
    <BlogList blogs={blogs} />  
  </div>  
)
```

```
    </div>
  );
}

export default App;
```

2. BlogList Component: **BlogList.jsx**

```
// src/components/BlogList.jsx
import React from 'react';
import BlogPost from './BlogPost'; // Import the BlogPost component
import PropTypes from 'prop-types';
import './BlogList.css';

function BlogList({ blogs }) {
  return (
    <div className="blog-list">
      {blogs.map((blog, index) => (
        <BlogPost
          key={index}
          title={blog.title}
          content={blog.content}
          author={blog.author}
        />
      ))}
    </div>
  );
}

BlogList.propTypes = {
  blogs: PropTypes.arrayOf(
    PropTypes.shape({
      title: PropTypes.string.isRequired,
      content: PropTypes.string.isRequired,
      author: PropTypes.string.isRequired,
    })
  ).isRequired,
};

export default BlogList;
```

3. BlogPost Component: **BlogPost.jsx**

```
// src/components/BlogPost.jsx
import React from 'react';
import PropTypes from 'prop-types';
import './BlogPost.css';

function BlogPost({ title, content, author }) {
  return (
    <div className="blog-post">
      <h2>{title}</h2>
      <p>{content}</p>
      <strong>Author: {author}</strong>
    </div>
  );
}

BlogPost.propTypes = {
  title: PropTypes.string.isRequired,
  content: PropTypes.string.isRequired,
  author: PropTypes.string.isRequired,
};

export default BlogPost;
```

4. BlogList CSS: **BlogList.css**

```
/* src/components/BlogList.css */
.blog-list {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-top: 20px;
}
```

5. App CSS (Optional):

```
/* src/App.css */
.App {
  font-family: 'Arial', sans-serif;
  background-color: #f7f7f7;
```

```
display: flex;
flex-direction: column;
align-items: center;
padding: 30px;
}

h1 {
color: #333;
font-size: 2rem;
margin-bottom: 30px;
font-weight: bold;
}

form {
margin-top: 30px;
display: flex;
flex-direction: column;
width: 100%;
max-width: 400px;
padding: 20px;
background-color: #fff;
border: 1px solid #ccc;
border-radius: 8px;
box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}

form div {
margin-bottom: 15px;
}

label {
font-size: 1.1rem;
font-weight: bold;
color: #333;
}

input,
textarea {
width: 100%;
padding: 10px;
font-size: 1rem;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
transition: border-color 0.3s ease;
}
```

```
input:focus,
textarea:focus {
  border-color: #4caf50;
  outline: none;
}

textarea {
  min-height: 100px;
}

button {
  padding: 12px;
  background-color: #4caf50;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 1rem;
  transition: background-color 0.3s ease, transform 0.3s ease;
}

button:hover {
  background-color: #45a049;
  transform: translateY(-2px);
}

button:active {
  transform: translateY(0);
}
```

Output: