# ReactJS Week 7 Exercises

## Exercise 4: ticketbookingapp

Objective: Implement conditional rendering in React based on login state.

App Name: ticketbookingapp

Functional Requirements:

- Guest user sees flight details.  
- Logged-in user can book tickets.  
- Conditional rendering based on login/logout.

### Code Summary:

App.js

import React, { useState } from 'react';  
import GuestPage from './GuestPage';  
import UserPage from './UserPage';  
  
function App() {  
 const [isLoggedIn, setIsLoggedIn] = useState(false);  
  
 const handleLogin = () => setIsLoggedIn(true);  
 const handleLogout = () => setIsLoggedIn(false);  
  
 return (  
 <div>  
 {isLoggedIn ? <UserPage onLogout={handleLogout} /> : <GuestPage onLogin={handleLogin} />}  
 </div>  
 );  
}  
  
export default App;

GuestPage.js

function GuestPage({ onLogin }) {  
 return (  
 <div>  
 <h2>Flight Details</h2>  
 <p>Only browsing available.</p>  
 <button onClick={onLogin}>Login</button>  
 </div>  
 );  
}  
  
export default GuestPage;

UserPage.js

function UserPage({ onLogout }) {  
 return (  
 <div>  
 <h2>Welcome, User!</h2>  
 <p>You can now book your flight.</p>  
 <button onClick={onLogout}>Logout</button>  
 </div>  
 );  
}  
  
export default UserPage;

### Screenshots with Explanation:

1. Guest View – The user is not logged in. They can only view flight details and a Login button.

A screenshot of a computer

AI-generated content may be incorrect.

2. User View – After clicking the Login button, the page shows a booking option and a Logout button.

A screenshot of a computer

AI-generated content may be incorrect.

## 📚 Exercise 5: bloggerapp

Objective: Use multiple ways of conditional rendering in a React app.

App Name: bloggerapp

Functional Requirements:

- Use components: BookDetails, BlogDetails, CourseDetails  
- Use: ternary, if-else, &&, element variables  
- Use map() with keys

### Code Summary:

App.js

import React, { useState } from 'react';  
import BookDetails from './BookDetails';  
import BlogDetails from './BlogDetails';  
import CourseDetails from './CourseDetails';  
  
function App() {  
 const [view, setView] = useState('book');  
  
 let element;  
 if (view === 'book') element = <BookDetails />;  
  
 return (  
 <div>  
 <button onClick={() => setView('book')}>Book</button>  
 <button onClick={() => setView('blog')}>Blog</button>  
 <button onClick={() => setView('course')}>Course</button>  
  
 {element}  
 {view === 'blog' ? <BlogDetails /> : null}  
 {view === 'course' && <CourseDetails />}  
 </div>  
 );  
}  
  
export default App;

BookDetails.js

function BookDetails() {  
 const books = ['Atomic Habits', 'React Explained'];  
 return (  
 <div>  
 <h3>Book List</h3>  
 <ul>  
 {books.map((book, index) => <li key={index}>{book}</li>)}  
 </ul>  
 </div>  
 );  
}  
  
export default BookDetails;

BlogDetails.js

function BlogDetails() {  
 return <h3>Welcome to Blogs</h3>;  
}  
  
export default BlogDetails;

CourseDetails.js

function CourseDetails() {  
 return <h3>ReactJS Course Information</h3>;  
}  
  
export default CourseDetails;

### Screenshots with Explanation:

1. Blog View – After clicking Blog, BlogDetails is rendered using ternary operator.

A screenshot of a computer

AI-generated content may be incorrect.

2. Course View – After clicking Course, CourseDetails is rendered using logical &&.

A screen shot of a computer

AI-generated content may be incorrect.

3. Book View – Default view shows BookDetails using an element variable.

A screenshot of a computer

AI-generated content may be incorrect.