## Explain about Conditional Rendering in React

Conditional Rendering in React refers to rendering different UI components based on logic or state conditions.  
It allows building dynamic interfaces that change based on user input or application status.  
Techniques used include ternary operators, logical AND (&&), if-else blocks, and switch-case statements in render functions.

## Define Element Variables

Element variables are variables that store JSX elements in React.  
They help organize complex rendering logic outside the return block for better readability.  
You can define them using const or let and use them conditionally in the render method.

## Explain how to prevent components from rendering

To prevent components from rendering, React provides techniques such as:  
1. Returning null from a component,  
2. Using conditional statements (if, &&, ternary) to skip rendering,  
3. In class components, using shouldComponentUpdate() to avoid re-rendering,  
4. In functional components, using useMemo or skipping rendering with state/props conditions.

# Ticket Booking Application Code

## App.jsx

import React, { useState } from 'react';  
import FlightList from './FlightList';  
import BookingForm from './BookingForm';  
import Login from './Login';  
  
function App() {  
 const [isLoggedIn, setIsLoggedIn] = useState(false);  
 const handleLogin = () => setIsLoggedIn(true);  
 const handleLogout = () => setIsLoggedIn(false);  
  
 return (  
 <div>  
 <h1>Flight Ticket Booking</h1>  
 {isLoggedIn ? (  
 <div>  
 <button onClick={handleLogout}>Logout</button>  
 <FlightList />  
 <BookingForm />  
 </div>  
 ) : (  
 <div>  
 <button onClick={handleLogin}>Login</button>  
 <FlightList />  
 </div>  
 )}  
 </div>  
 );  
}  
  
export default App;

## FlightList.jsx

import React from 'react';  
  
const flights = [  
 { id: 1, origin: 'New York', destination: 'London', time: '10:00 AM' },  
 { id: 2, origin: 'Paris', destination: 'Tokyo', time: '1:00 PM' },  
 { id: 3, origin: 'Berlin', destination: 'Dubai', time: '6:00 PM' },  
];  
  
function FlightList() {  
 return (  
 <div>  
 <h2>Available Flights</h2>  
 <ul>  
 {flights.map(flight => (  
 <li key={flight.id}>  
 {flight.origin} to {flight.destination} at {flight.time}  
 </li>  
 ))}  
 </ul>  
 </div>  
 );  
}  
  
export default FlightList;

## BookingForm.jsx

import React, { useState } from 'react';  
  
function BookingForm() {  
 const [name, setName] = useState('');  
 const [flightId, setFlightId] = useState('');  
  
 const handleSubmit = (e) => {  
 e.preventDefault();  
 alert(\`Ticket booked for \${name} on flight ID \${flightId}!\`);  
 };  
  
 return (  
 <form onSubmit={handleSubmit}>  
 <h2>Book a Flight</h2>  
 <input  
 type="text"  
 placeholder="Enter your name"  
 value={name}  
 onChange={(e) => setName(e.target.value)}  
 />  
 <input  
 type="text"  
 placeholder="Enter Flight ID"  
 value={flightId}  
 onChange={(e) => setFlightId(e.target.value)}  
 />  
 <button type="submit">Book Ticket</button>  
 </form>  
 );  
}  
  
export default BookingForm;

## Login.jsx

import React from 'react';  
  
function Login({ onLogin }) {  
 return (  
 <div>  
 <h2>Please Log In</h2>  
 <button onClick={onLogin}>Login</button>  
 </div>  
 );  
}  
  
export default Login;

## Output



