**Delete Movie by Admin Using Kafka:**

In MovieBooking application after logged in as admin, admin can only delete the movie. For that we used kafka to delete the movie.

// --- DELETE MOVIE  
@DeleteMapping("/{movieName}/delete")@SecurityRequirement(name = "Bearer Authentication")  
@Operation(summary = "delete a movie(Admin Only)")  
@PreAuthorize("hasRole('ADMIN')")  
public ResponseEntity<String> deleteMovie(@PathVariable String movieName){  
List<Movie> availableMovies = movieService.findByMovieName(movieName);  
if(availableMovies.isEmpty()){  
throw new MoviesNotFound("No movies Available with moviename "+ movieName);  
}  
else {  
movieService.deleteByMovieName(movieName);  
**kafkaTemplate.send(topic.name(),"Movie Deleted by the Admin. "+movieName+" is now not available");**return new ResponseEntity<>("Movie deleted successfully",HttpStatus.OK);  
}  
  
}

**React Components:**

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions but work in isolation and return HTML.

Components come in two types, Class components and Function components.

**Class components in React:**

React **class components** are the bread and butter of most modern web apps built in ReactJS. These components are simple classes (made up of multiple functions that add functionality to the application). All **class components** are child classes for the Component class of ReactJS.

**Example for Class Component:**

/\* Register.jsx \*/  
import React from "react";  
import loginImg from "../../login.svg";  
  
export class Register extends React.Component {  
constructor(props) {  
super(props);  
}  
  
render() {  
return (  
<div className="base-container" ref={this.props.containerRef}>  
<div className="header">Register</div>  
<div className="content">  
<div className="image">  
<img src={loginImg} />  
</div>  
<div className="form">  
<div className="form-group">  
<label htmlFor="username">Username</label>  
<input type="text" name="username" placeholder="username" />  
</div>  
<div className="form-group">  
<label htmlFor="email">Email</label>  
<input type="text" name="email" placeholder="email" />  
</div>  
<div className="form-group">  
<label htmlFor="password">Password</label>  
<input type="text" name="password" placeholder="password" />  
</div>  
</div>  
</div>  
<div className="footer">  
<button type="button" className="btn">  
Register  
</button>  
</div>  
</div>  
);  
}  
}

**Function Component:**

Here is the same example as above but created using a Function component instead.

A Function component also returns HTML and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand.

Example of Calling the function using component call:

import React from 'react';  
import ReactDOM from 'react-dom/client';  
function Comp() {  
 return (<h1> As usual we can call the function using component call</h1>);  
}  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<Comp />);