<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Survey Program</title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.14.0/umd/react.development.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.14.0/umd/react-dom.development.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/babel.min.js"></script>

</head>

<body>

<div id="root"></div>

<script type="text/babel">

function SurveyProgram() {

const [scores, setScores] = React.useState([]);

const [showModal, setShowModal] = React.useState(false);

const handleScoreClick = (score) => {

if (score === 5) {

setShowModal(true);

}

setScores((prevScores) => [...prevScores, score]);

};

const handleCloseModal = () => setShowModal(false);

return (

<div className="survey-container">

<h1>Telephone Company Provider Survey</h1>

<p>Rate your satisfaction with the service:</p>

<div className="score-buttons">

<button className="score-btn" onClick={() => handleScoreClick(1)}>1</button>

<button className="score-btn" onClick={() => handleScoreClick(2)}>2</button>

<button className="score-btn" onClick={() => handleScoreClick(3)}>3</button>

<button className="score-btn" onClick={() => handleScoreClick(4)}>4</button>

<button className="score-btn" onClick={() => handleScoreClick(5)}>5</button>

</div>

<h2>Scores:</h2>

{scores.length === 0 ? (

<p>No scores yet.</p>

) : (

<ul>

{scores.map((score, index) => (

<li key={index}>{score}</li>

))}

</ul>

)}

{scores.includes(5) && (

<div className="emoji">

<span role="img" aria-label="Party emoji">🎉</span>

</div>

)}

{showModal && (

<div className="modal">

<div className="modal-content">

<h3>Thank you!</h3>

<p>Your feedback is greatly appreciated.</p>

<button className="close-btn" onClick={handleCloseModal}>Close</button>

</div>

</div>

)}

</div>

);

}

ReactDOM.render(

<React.StrictMode>

<SurveyProgram />

</React.StrictMode>,

document.getElementById('root')

);

</script>

<style>

/\* Body styling with purple background \*/

body {

background-color: #9b59b6; /\* Purple background \*/

font-family: Arial, sans-serif;

color: white;

margin: 0;

padding: 0;

}

.survey-container {

background-color: #fff;

color: #333;

padding: 20px;

border-radius: 10px;

width: 80%;

max-width: 600px;

margin: 50px auto;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

text-align: center;

}

h1 {

color: #9b59b6; /\* Purple color for title \*/

}

.score-buttons {

display: flex;

justify-content: center;

gap: 10px;

margin-bottom: 20px;

}

.score-btn {

padding: 10px 20px;

font-size: 16px;

background-color: #9b59b6; /\* Purple button \*/

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

.score-btn:hover {

background-color: #8e44ad; /\* Darker purple on hover \*/

}

.emoji {

font-size: 60px;

margin-top: 20px;

animation: jump 1s ease infinite;

}

/\* Animation for the jumping emoji \*/

@keyframes jump {

0% {

transform: translateY(0);

}

25% {

transform: translateY(-20px);

}

50% {

transform: translateY(0);

}

75% {

transform: translateY(-10px);

}

100% {

transform: translateY(0);

}

}

.modal {

position: fixed;

top: 0;

left: 0;

right: 0;

bottom: 0;

background-color: rgba(0, 0, 0, 0.5);

display: flex;

justify-content: center;

align-items: center;

}

.modal-content {

background-color: #fff;

padding: 20px;

border-radius: 5px;

text-align: center;

width: 300px;

}

.close-btn {

padding: 10px 20px;

font-size: 16px;

background-color: #9b59b6; /\* Purple button \*/

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

margin-top: 10px;

}

.close-btn:hover {

background-color: #8e44ad; /\* Darker purple on hover \*/

}

h2, p, ul {

font-family: Arial, sans-serif;

}

ul {

list-style-type: none;

padding: 0;

}

li {

margin: 5px 0;

}

</style>

</body>

</html>