|  |  |  |  |
| --- | --- | --- | --- |
|  | **Course Name: Advanced Web Technology** | **EXPERIMENT NO. 8** | |
| **Course Code: 20CP314P**  **Faculty: Komal Singh** | **Branch: CSE** | **Semester: VI** |
| **Submitted by: Patel Vraj ChetanKumar**  **Roll no: 21BCP362** | | | |

Objective: Working with forms in React

Experiment 8:

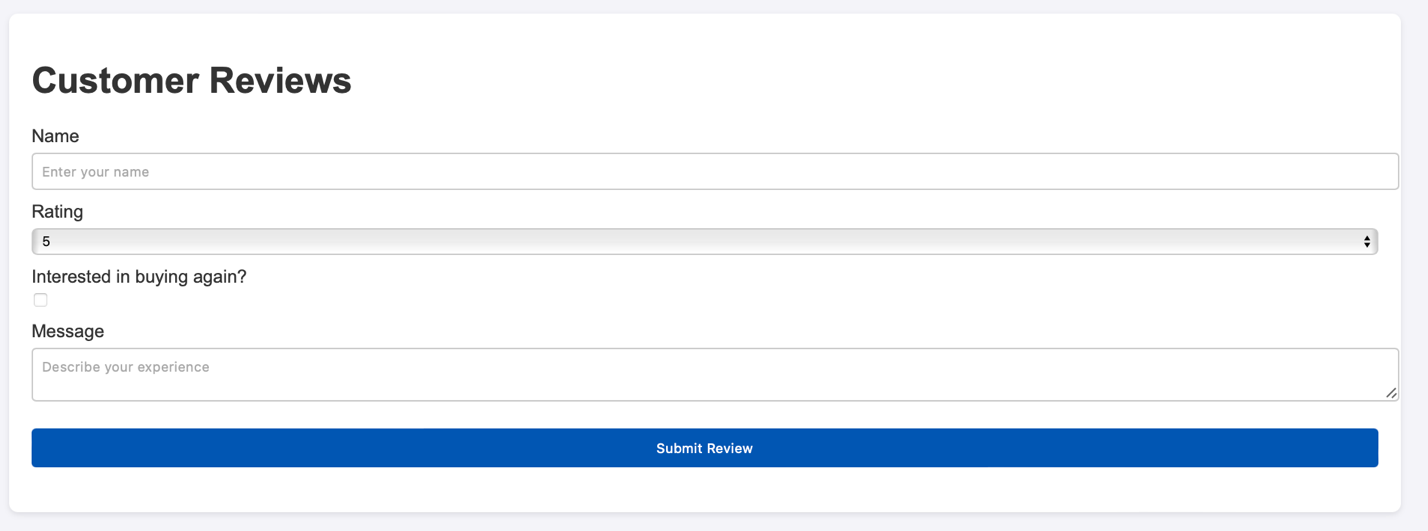
1. Create a customer review form using react forms for the commercial furniture web-application with multiple fields such as name of the customer(text), the ratings for the customer experience out of 5(a drop-down menu), whether interested in buying again in future (check-box yes or no) and message from the customer(text-area) also ensure that the message should be less than 25 characters.
2. Create a react form having fields like name(text), division (drop down menu) and message(text-area) using the useRef hook.

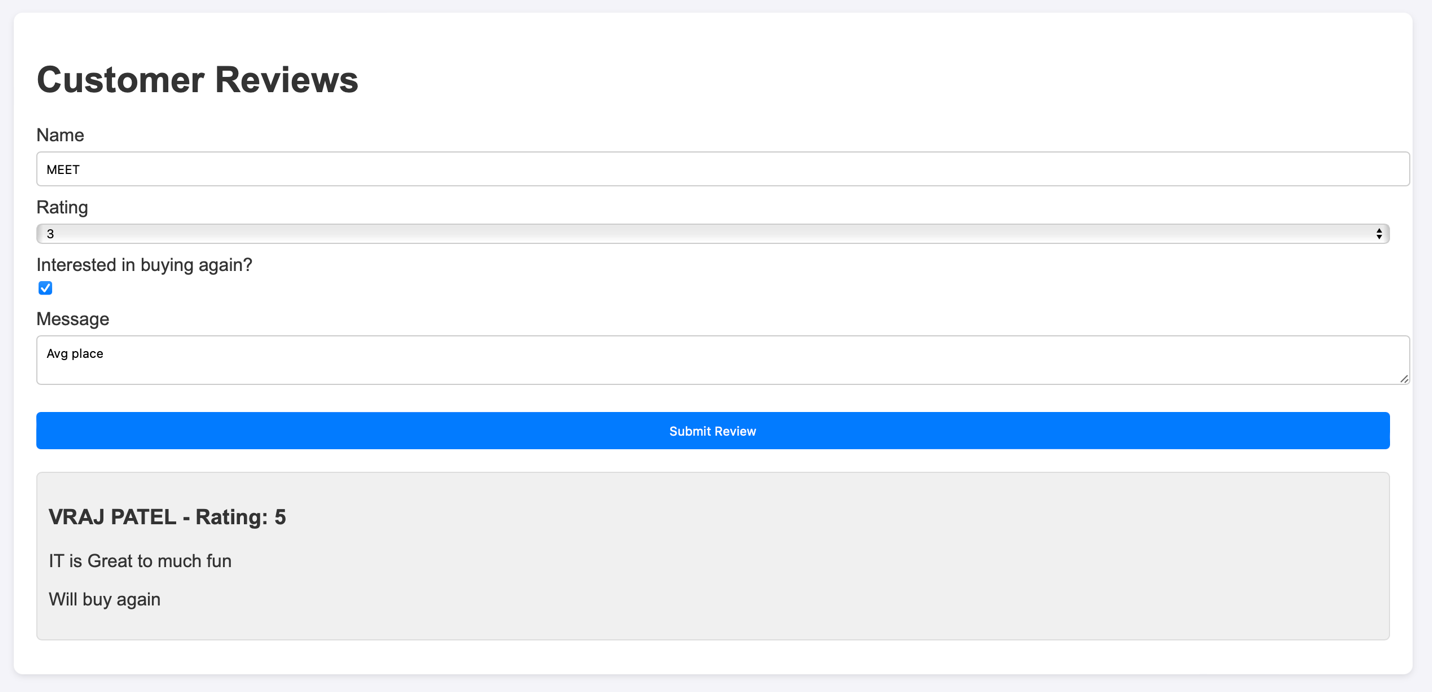
Note: Please include snapshots of all commands, terminal sessions, localhost outputs, and log files in your documentation with necessary steps.

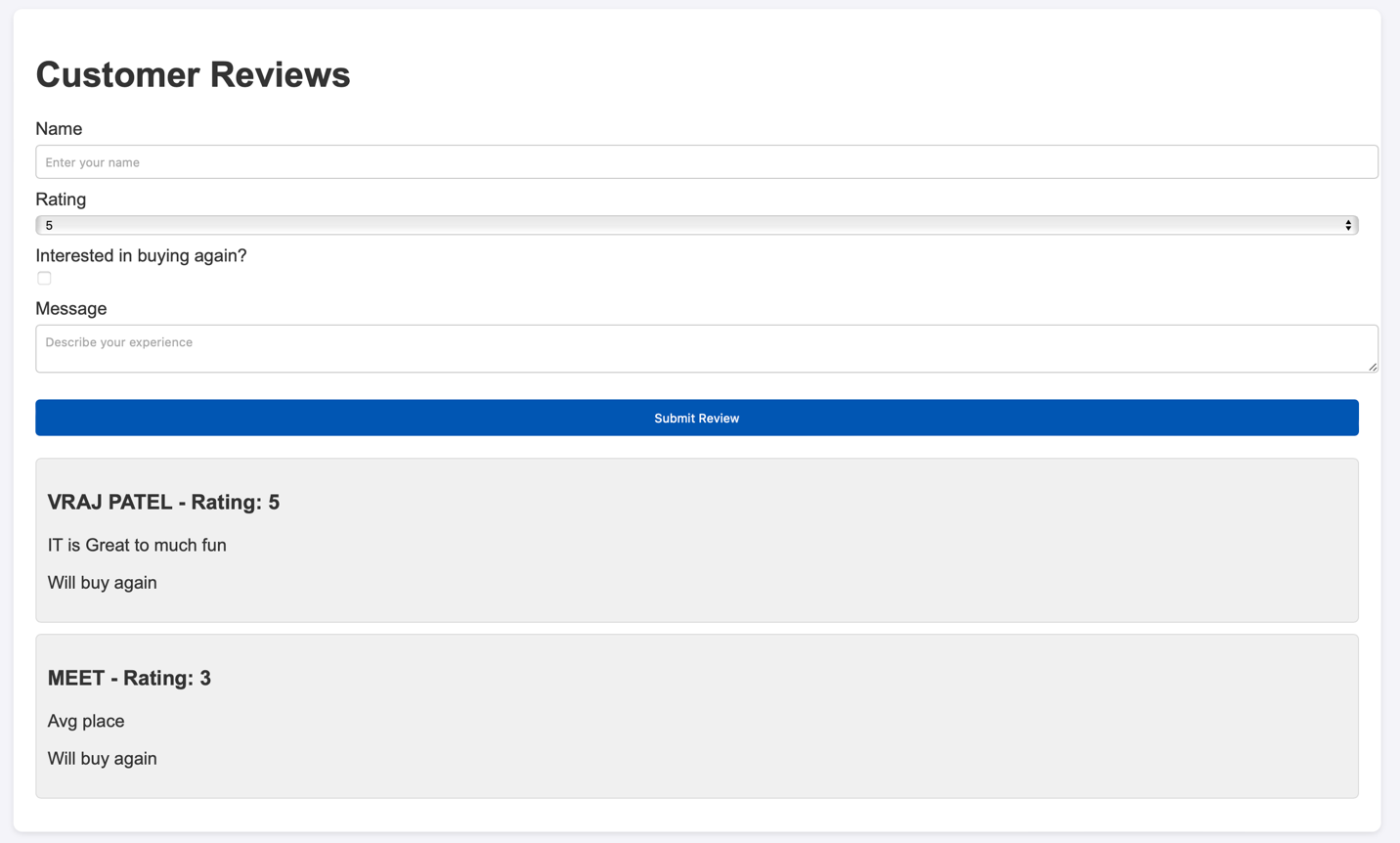
**8a: Controlled Form with Dynamic Review Display**

We'll create a form where customers can submit reviews and see their reviews displayed immediately below the form.

|  |
| --- |
| // src/pages/CustomerReview.js  import React, { useState } from 'react';  import './PageStyles.css';  function CustomerReview() {  const [review, setReview] = useState({  name: '',  rating: '5',  willBuyAgain: false,  message: ''  });  const [reviews, setReviews] = useState([]);  const handleInputChange = (event) => {  const { name, value, type, checked } = event.target;  setReview({  ...review,  [name]: type === 'checkbox' ? checked : value  });  };  const handleSubmit = (event) => {  event.preventDefault();  // Add the review to the reviews array  setReviews([...reviews, review]);  // Reset the form  setReview({  name: '',  rating: '5',  willBuyAgain: false,  message: ''  });  };  return (  <div className="container review-container">  <h1>Customer Reviews</h1>  <form className="review-form" onSubmit={handleSubmit}>  <label htmlFor="name">Name</label>  <input  id="name"  name="name"  type="text"  placeholder="Enter your name"  value={review.name}  onChange={handleInputChange}  />  <label htmlFor="rating">Rating</label>  <select  id="rating"  name="rating"  value={review.rating}  onChange={handleInputChange}  >  <option value="5">5</option>  <option value="4">4</option>  <option value="3">3</option>  <option value="2">2</option>  <option value="1">1</option>  </select>  <label>  Interested in buying again?  <input  name="willBuyAgain"  type="checkbox"  checked={review.willBuyAgain}  onChange={handleInputChange}  />  </label>  <label htmlFor="message">Message</label>  <textarea  id="message"  name="message"  placeholder="Describe your experience"  maxLength="25"  value={review.message}  onChange={handleInputChange}  />  <button type="submit">Submit Review</button>  </form>  <div className="reviews-display">  {reviews.map((item, index) => (  <div key={index} className="review-card">  <h3>{item.name} - Rating: {item.rating}</h3>  <p>{item.message}</p>  <p>{item.willBuyAgain ? 'Will buy again' : 'Not sure about buying again'}</p>  </div>  ))}  </div>  </div>  );  }  export default CustomerReview; |





Dynamic Form

**Experiment 8b: Uncontrolled Form Using useRef Hook**

|  |
| --- |
| // src/pages/UncontrolledForm.js (You can integrate this into CustomerReview.js or as a separate component)  import React, { useRef } from 'react';  import './PageStyles.css';  function UncontrolledForm() {  const nameRef = useRef(null);  const divisionRef = useRef(null);  const messageRef = useRef(null);  const reviewsRef = useRef([]);  const handleSubmit = (event) => {  event.preventDefault();  const newReview = {  name: nameRef.current.value,  division: divisionRef.current.value,  message: messageRef.current.value  };  // Update reviews display without state management  reviewsRef.current.push(newReview);  displayReviews();    // Optionally, reset form fields here if needed  event.target.reset();  };  const displayReviews = () => {  const reviewsContainer = document.querySelector('.uncontrolled-reviews-display');  reviewsContainer.innerHTML = ''; // Clear previous content  reviewsRef.current.forEach(review => {  const reviewDiv = document.createElement('div');  reviewDiv.className = 'review-card';  reviewDiv.innerHTML = `  <h3>${review.name} - Division: ${review.division}</h3>  <p>${review.message}</p>  `;  reviewsContainer.appendChild(reviewDiv);  });  };  return (  <div className="container review-container">  <h1>Uncontrolled Customer Reviews</h1>  <form className="review-form" onSubmit={handleSubmit}>  <label htmlFor="un\_name">Name</label>  <input id="un\_name" ref={nameRef} type="text" placeholder="Enter your name" />    <label htmlFor="un\_division">Division</label>  <select id="un\_division" ref={divisionRef}>  <option value="Sales">Sales</option>  <option value="Marketing">Marketing</option>  <option value="Development">Development</option>  <option value="Human Resources">Human Resources</option>  </select>  <label htmlFor="un\_message">Message</label>  <textarea id="un\_message" ref={messageRef} placeholder="Your message" />  <button type="submit">Submit Review</button>  </form>  <div className="uncontrolled-reviews-display"></div>  </div>  );  }  export default UncontrolledForm; |

