

TASK-17

17. Create a Controlled Form:

- Build a `FeedbackForm.jsx` that captures name, email, and feedback message.

Final File Structure:

```
/src
  /components
    | FeedbackForm.jsx
  /styles
    | FeedbackForm.css
  App.js
```

1. Create FeedbackForm.jsx:

```
// /src/components/FeedbackForm.jsx

import React, { Component } from 'react';
import './styles/FeedbackForm.css'; // Import the CSS file for styling

class FeedbackForm extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: "",
      email: "",
      feedback: "",
    };
  }

  // Bind the methods to 'this'
  this.handleChange = this.handleChange.bind(this);
  this.handleSubmit = this.handleSubmit.bind(this);
}

// Handle changes in form inputs and update state
handleChange(event) {
  const { name, value } = event.target;
  this.setState({ [name]: value }); // Dynamically set the state for name, email, feedback
}
```

```
// Handle form submission
handleSubmit(event) {
  event.preventDefault();
  // Display the form data (in a real-world app, send it to a server or API)
  alert(` Feedback submitted:\nName: ${this.state.name}\nEmail:
${this.state.email}\nFeedback: ${this.state.feedback}` );
  this.setState({
    name: "",
    email: "",
    feedback: "",
  }); // Clear the form fields after submission
}

render() {
  return (
    <div className="feedback-form">
      <h2>Student Feedback Form</h2>
      <form onSubmit={this.handleSubmit}>
        <div className="form-group">
          <label htmlFor="name">Name:</label>
          <input
            type="text"
            id="name"
            name="name"
            value={this.state.name}
            onChange={this.handleChange}
            required
          />
        </div>

        <div className="form-group">
          <label htmlFor="email">Email:</label>
          <input
            type="email"
            id="email"
            name="email"
            value={this.state.email}
            onChange={this.handleChange}
            required
          />
        </div>

        <div className="form-group">
          <label htmlFor="feedback">Feedback:</label>
          <textarea
            id="feedback"

```

```

    name="feedback"
    value={this.state.feedback}
    onChange={this.handleInputChange}
    rows="4"
    required
  ></textarea>
</div>

<div className="form-group">
  <button type="submit">Submit Feedback</button>
</div>
</form>

  {/* Centered Design By Het Dabhi text */}
  <div className="centered-footer">
    <p>@Design By Het Dabhi</p>
  </div>
</div>
);
}
}

export default FeedbackForm;

```

2. Create FeedbackForm.css (Optional Styling):

```

/* /src/styles/FeedbackForm.css */

/* General styling for the feedback form container */
.feedback-form {
  width: 100%;
  max-width: 600px; /* Limits the width to 600px */
  margin: 50px auto; /* Centers the form on the page */
  padding: 20px;
  background-color: #f9f9f9; /* Light background color for the form */
  border-radius: 8px; /* Rounded corners */
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Soft shadow around the form */
}

/* Styling for the form header */
h2 {
  font-size: 1.8em;
  margin-bottom: 20px; /* Space below the header */
  text-align: center; /* Center the header text */

```

```

}

/* Styling for each form field (input, textarea) */
.form-group {
  margin-bottom: 20px; /* Space between form fields */
}

label {
  display: block;
  font-size: 1.2em;
  margin-bottom: 5px; /* Space below the label */
}

/* Input fields (text and email) */
input[type="text"],
input[type="email"],
textarea {
  width: 100%;
  padding: 10px; /* Padding inside the input fields */
  font-size: 1.2em; /* Larger font size */
  border: 1px solid #ccc; /* Light border color */
  border-radius: 4px; /* Rounded corners for the input fields */
}

/* Textarea field styling */
textarea {
  resize: vertical; /* Allow vertical resizing */
}

/* Styling for the submit button */
button {
  width: 100%;
  padding: 10px; /* Button padding */
  font-size: 1.2em; /* Button text size */
  background-color: #4caf50; /* Green background for the button */
  color: white; /* White text color */
  border: none; /* No border for the button */
  border-radius: 4px; /* Rounded corners for the button */
  cursor: pointer; /* Pointer cursor when hovering over the button */
}

button:hover {
  background-color: #45a049; /* Darker green when hovering */
}

/* Styling for the centered footer text */
.centered-footer {

```

```
width: 100%;  
text-align: center; /* Center the text */  
margin-top: 20px; /* Space from the form */  
}  
  
.centered-footer p {  
margin: 0;  
font-size: 1.2em; /* Font size for the centered text */  
font-style: italic; /* Italicize the text */  
}
```

3. Integrate FeedbackForm.jsx in App.js:

```
// /src/App.js  
  
import React from 'react';  
import FeedbackForm from './components/FeedbackForm';  
  
function App() {  
  return (  
    <div className="App">  
      <FeedbackForm />  
    </div>  
  );  
}  
  
export default App;
```

4. App.css: (For Styling)

```
/* src/App.css */  
.App {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  background-color: #f7f7f7;  
  font-family: Arial, sans-serif;  
  height: 100vh;  
}  
  
.App-header {
```

```
background-color: #4CAF50;
color: white;
width: 100%;
padding: 20px;
text-align: center;
font-size: 2em;
}

.content-wrapper {
display: flex;
justify-content: center;
width: 100%;
padding: 20px;
flex-grow: 1; /* Ensures the content takes all available space */
}

/* App-level footer */
.App-footer {
width: 100%;
background-color: #333;
color: white;
text-align: center;
padding: 10px 0;
position: relative;
bottom: 0;
margin-top: auto; /* Pushes the footer to the bottom */
}

.App-footer p {
margin: 0;
font-size: 1em;
}
```

Output:

Student Feedback Form

Name:

Het Dabhi

Email:

hetdabhi@gmail.com

Feedback:

Thank You !!

Submit Feedback

@Design By Het Dabhi