Controlled Form Application Data Flow

1. Introduction

This document provides a detailed Data Flow Diagram (DFD) representation of the Controlled Form React application.

It explains how controlled form inputs work, how state is managed, and how UI updates dynamically.

2. System Overview

The React application consists of the following key components:

- App.js: The main file that initializes the application and renders child components.
- components/
  - ControlledForm.js: Manages user input in a controlled manner using useState.
  - ToggleText.js: Toggles the visibility of text on button click.

3. Data Flow Diagram (DFD)

Level 0 (Context Diagram)

At the highest level, the system consists of external users interacting with form elements and buttons.

External Entities:

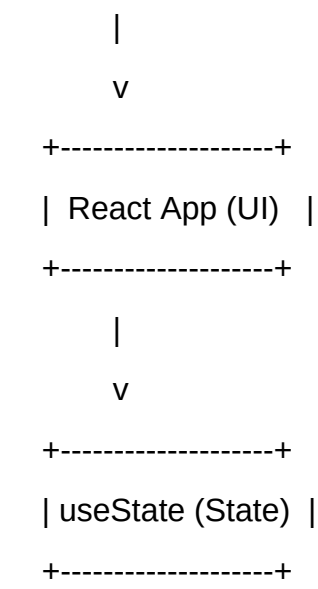
- User: Provides input and interacts with UI components.

Processes:

- React Application: Handles user input, state updates, and dynamic UI rendering.

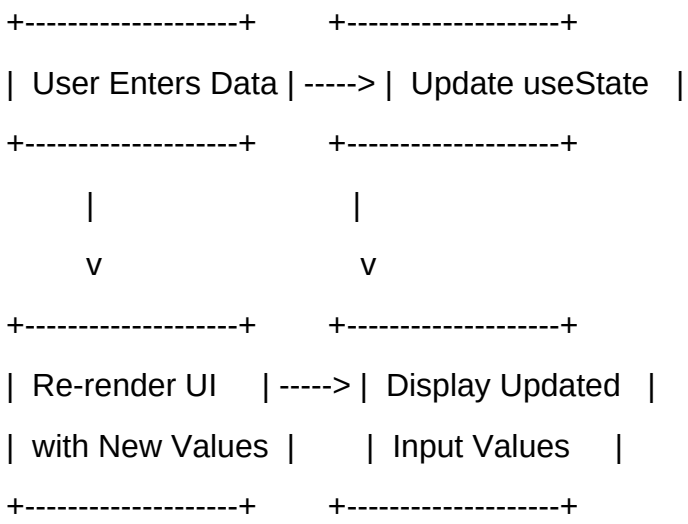
Data Stores:

- useState Hook: Stores form input values and toggle state.



Level 1 DFD (Controlled Form Process)

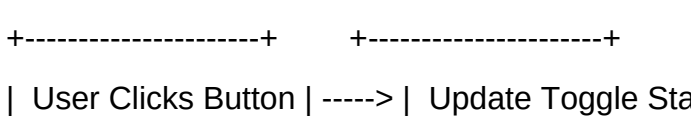
This level breaks down the controlled form data flow.

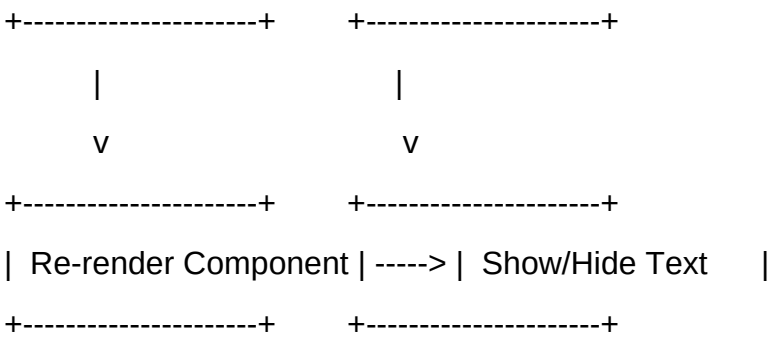


Process Explanation:

1. The user enters text into the input fields.
2. The onChange event updates the corresponding useState variable.
3. React detects the state change and re-renders the component with updated values.

Level 2 DFD (ToggleText Component Process)





Process Explanation:

1. The user clicks the button to toggle text visibility.
2. The onClick event updates the state variable (true/false).
3. The component re-renders, showing or hiding the text accordingly.

4. Explanation of Data Flow

1. The user interacts with input fields in ControlledForm.
2. The component manages state using useState and updates dynamically.
3. The user clicks the toggle button in ToggleText to show/hide content.
4. The application re-renders based on user interactions.

5. Conclusion

This document outlines the detailed data flow in a React-based controlled form and toggle component system.

The application dynamically manages form state and text visibility using React state management.