

Animesh Parab 88 T2-T21

Assignment 7B

Aim:- WAP to implement the concept of React Hooks.

Lo Mapped :- Lo 5

Theory:-

This assignment focuses on implementing the concept of React Hooks to handle user input through a text input box. React Hooks, specifically the `useState` Hook, will be employed to create a dynamic and responsive user interface. The theory section will cover the principles of React Hooks and how they enable interactive input handling in React applications.

What are React Hooks?

- React Hooks are functions provided by the React library to allow functional components to manage state and side-effects.
- They were introduced in React 16.8 to eliminate the need for class components in many cases, enabling state management in functional components.

useState Hook:

- The `useState` Hook is used to create and manage state within a functional component.
- It takes an initial state value as an argument and returns an array with two elements: the current state and a function to update it.
- The initial state is set when the component is first rendered.

Handling User Input:

- React Hooks can be used to capture and manage user input in real-time.
- To capture user input, create a state variable using `useState`.
- Attach an `onChange` event handler to the input element to update the state variable as the user types.
- The input element's `value` attribute should be set to the state variable to reflect the current input value.

Code :-

```
App.js
File Edit View
import React, { useState } from 'react';
function TextBoxExample() {
  const [text, setText] = useState('');
  const handleInputChange = (e) => {
    setText(e.target.value);
  };
  return (
    <div>
      <h1> Text Box </h1>
      <input
        type="text"
        placeholder="Enter text"
        value={text}
        onChange={handleInputChange}
      />
      <p>You entered: {text}</p>
    </div>
  );
}
export default TextBoxExample;
Ln 20, Col 9 100% Unix (LF) UTF-8
```

Output :-



Text Box

You entered: Animesh

Conclusion :-

React Hooks, particularly the `useState` Hook, have transformed the way developers handle user input in React applications. They offer a clean and efficient solution to capture and respond to user interactions, making it easier to create responsive and interactive user interfaces. By incorporating React Hooks into the assignment, students can gain valuable hands-on experience with state management and user input handling, essential skills for building modern web applications using React.