NAVYA NARAYAN PANICKER

WEEK 2

DAY 3: 1/10/25[WEDNESDAY]

TASK : Add ability to add tasks; manage state; fetch effects if needed (dummy), Create form using MUI TextFields and Buttons; validate required inputs / length / number etc.

QUESTIONS/REFLECTIONS:

1. Walk through flow: user types → state updates → component re-renders → effect runs (if any).

   User event occurs: the event handlers runs and calls this.setState({ value: newValue })

   State update is scheduled: react marks the component to be updated.multiple setstate calls maybe done.

   Render: React calls render() with the new state to produce the new VDOM.

   Commit phase: the DOM updates are applied.

   Cleanup: If an update causes child components to unmount,
   their componentWillUnmount runs synchronously during the commit.

2. How useEffect works: dependencies, cleanup, initial render.

   The useeffect in react runs the side effect code after a component renders like getting the data or updating the DOM.
   It will accept a dependency array, if it is empty the effect will run only once after the initial render. If the dependencies are listed then it will run those values change.
   It can also return a cleanup function which runs before the effect reexecutes.

3. What pitfalls exist (e.g. stale closures, infinite loops)?
   If u forget the dependency array will make the useeffect run after every render.
   This leads to slowing down or cause infinite loops.
   It make the eseeffect use old values which leads to bugs.
   Updating state inside the use-effect without a condition makes an infinite loop.
   Not cleaning up things like timers or event listeners can cause memory leaks.

4. What is a controlled vs uncontrolled component?
   controlled is when the element whos values are managed by the react state so react knows and controls the input values.

   Uncontrolled components is when the element whos values are not managed by react state so react does not know the input valies and cannot control it.

5. Describe event handling in forms (onChange, onSubmit).

Onchange is an event when the user changes something react will immediately update it and set the new values.

Onsubmit runs when the user submits the form, letting the app process the data, validates the data and prevents it from default reload.

6. How does MUI help: theming, form helpers, error display?
   Theming :
   It uses themeprovider to let the developers set different color, font, and other details.

   Form helpers:
   ready-made form components like TextField, Checkbox, and Select
   it handles common logic and how to access it.

   Error display:
   It has built in props to check the error handling
   It will validate the error messages in a easier and more efficient way.