**REACT**

**WEEK-6 HANDSON-4 EXPLANATION**

**Clarifying need and advantages of component lifecycle**

Component lifecycle in React describes the phases through which a component passes in its lifetime.

It talks about creation, updating, and unmounting.

The lifecycle methods allow developers to execute some code at specific points of the life cycle of a component.

Helps manage side effects such as data-fetching, subscriptions, DOM updates or cleanup effectively.

Allows performance optimization of what runs and when.

Pretty useful in debugging and for component behavior monitoring.

**Identify various lifecycle hook methods**

**For class components:**

constructor() – Initialization

render() – UI rendering

componentDidMount() – After first render

componentDidUpdate(prevProps, prevState) – After updates

componentWillUnmount() – Cleanup before removal

shouldComponentUpdate(nextProps, nextState) – Control re-rendering

componentDidCatch(error, info) – For graceful error handling

**For Functional Components (React Hooks):**

useEffect() – As componentDidMount, componentDidUpdate, componentWillUnmount based on dependencies

useLayoutEffect() – Similar to useEffect but fires synchronously after render

useRef() – Used for keeping reference across renders

useState() – Holds state of a component

**Enumerate the steps involved in rendering a component**

**Mounting phase**

constructor()

static getDerivedStateFromProps()

render()

componentDidMount()

**Updating phase**

static getDerivedStateFromProps()

shouldComponentUpdate()

render()

getSnapshotBeforeUpdate()

componentDidUpdate()

**Unmounting phase**

componentWillUnmount()

**Error handling (optional)**

componentDidCatch()