**React**

**Day 27th**

**State**

State is a plain JS object used by react to represent an information about the component current situation.

**What is state object in react??**

React components has a built in state object. The state object is where you store property values that belong to the component. When the state object changes the component re-renders.

State can hold any JS value, including objects. But you should not change objects that you hold in the react state directly. Instead, when you want to update an object, you need to create a new one (or make a copy of an existing one) and then set the state to use that copy.

A state us an object that holds data and information related to a react component. It can be used to store, manage and update data within application, which in turn allows for dynamic changes to user interfaces. For example if a button needs to change its text based I the user input then this would be done by updating the state within new values.

**What is setState in React**

setState() enqueues changes to the component state and tells react that this component and its children need to be re-rendered with the updated state. This is primary method you use to update the user interface in response to event handlers and server responses.

setState() allows you to change state in class based component.

📝 setState() does not always update the component immediately instead it may defer from the update until later for better performance. It is asynchronous in nature.

**Syntax:**

**setState(updater,[callback])**

**updater:**

A function with the signature: (state, props) => stateChange.

State is the current, up to date component state and the function returns the updated state

**Callback**

An optional callback function that is guaranteed to run after the state has been updated.