***Module (React Js) – 7***

**1). What is Redux?**

**Ans.** Redux is an open-source JavaScript library for controlling an application's state. While it can be utilized with different frameworks or even stand-alone JavaScript apps, it is most frequently connected with web development through the use of libraries like React.

Redux is primarily meant to give JavaScript applications a reliable state container. It facilitates centralized, predictable state management of an application and enforces a unidirectional data flow.

**2). What is Redux Thunk used for?**

**Ans.** The middleware for the Redux state management library is called Redux Thunk.

It enables you to build function-returning action creators as opposed to simple action objects.

In Redux apps, this is very helpful for managing asynchronous tasks like making API requests.

Example:

// Standard action creator returning an object

const fetchDataRequest = () => ({

type: 'FETCH\_DATA\_REQUEST'

});

//Fetch Data Request

**3). What is Pure Component? When to use Pure Component over Component?**

**Ans.** Pure Component is a class component provided by React that extends React.

PureComponent instead of React.Component. The key feature of Pure Components is that they automatically implement the shouldComponentUpdate lifecycle method with a shallow prop and state comparison. This means that a Pure Component will only re-render if the props or state change at a level deeper than a shallow comparison.

// Regular React Component

class MyComponent extends React.Component {

shouldComponentUpdate(nextProps, nextState) {

// Custom logic to determine whether to re-render

return true;

}

render() {

// Render logic

}

}

**4). What is the second argument that can optionally be passed tosetState and what is its purpose?**

**Ans.** In React, the setState function can take an optional second argument, which is a callback function. The purpose of this callback function is to perform additional logic or side effects after the state has been successfully updated.

The setState function is asynchronous, and React batches state updates for performance reasons. This means that if you want to perform some action after the state has been updated, you should use the callback function. The callback is executed once the state has been applied and the component has been re-rendered.