React- Applying Redux

REACT

Module- 7

**(React – Applying Redux)**

**Submitted To:-**

**Rajesh Nagar Sir**

**Submitted By:-**

**Ankita Sahu**

Module-7) React - Applying Redux

**1. What is Redux?**

**Ans:-** Redux is the most powerful state management tool in the React. Js library, and it's frequently used by Frontend Engineers to build complex applications. With the help of Redux, we can easily manage the state of our React applications, which helps us to develop and maintain complex applications.

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

**Ans:-** Redux Thunk middleware allows you to write action creators that return a function instead of an action. The thunk can be used to delay the dispatch of an action, or to dispatch only if a certain condition is met. The inner function receives the store methods dispatch and getState as parameters.

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

**Ans:-** Components are like the building blocks of our app. There are regular components and pure components. Pure components, whether they're classes or functions, are special ones that aim to make your app run faster by avoiding unnecessary updates/re-renders.

* Pure components are specifically designed for performance optimization by minimizing unnecessary renders through automatic prop comparison. Regular components, both functional and class-based, might re-render even if props or state don't change, leading to potential performance issues.

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

**Ans:-** The setState function is used to update the state of a component.

* It takes up to two arguments:

1. The first argument is the new state or a function that returns the new state.
2. The second argument is an optional callback function.

The purpose of the second argument, the callback function, is to execute code after the state has been updated and the component has re-rendered. This is useful because setState is asynchronous, meaning that state changes and re-renders might not happen immediately. The callback ensures that you can perform actions that depend on the updated state and the re-rendered component.

Example :-



**5. Create a Table and Search data from table using React Js?**

**Ans:- See the Folder\* Ques\_5.**

**6. Create Login registration with CRUD Application using API (Redux).**

**Ans:- See the Folder\* Ques\_6.**