Congratulations! You were able to apply the core concepts of the Redux framework by implementing an application using the Redux library.

By completing this lesson, you are now able to:

* Install the redux library into your project using npm install redux.
* Import the createStore() helper function from the 'redux' library.
* Create a store object that holds the entire state of your Redux application using createStore().
* Get the current state of the store using store.getState().
* Dispatch actions to the store using store.dispatch(action).
* Create action creators to reduce the repetitive creation of action objects.
* Register a change listener function to respond to changes to the store using store.subscribe(listener).
* Recognize the pattern for connecting Redux to any user interface.
* Implement a Redux application using either the HTML DOM API or React.

Throughout this lesson, you may have thought to yourself that Redux adds a lot of unnecessary complexity to these simple applications. We implemented Redux in a very basic way, which is useful for learning but not how it’s done in the real world.

Redux shines when it is used in applications with many features and a lot of data where having a centralized store to keep it all organized is advantageous. In the next lesson, you will learn how to build and organize Redux applications with complex state.