React Task 6

**Deadline: 23.05**

**Summary**

Based on your previous tasks integrate Redux for state management (you need both react and react-redux dependencies).

Store all the application data into Redux state (e.g. cards data, user data, authorization status).

**Implementation notes**

**1)** Create an additional folder in your app ( store, redux or something else). This folder will contain the store of your whole application(createStore), all the required reducers for your app (if you have more than 1 reducer, don’t forget to [combine](https://redux.js.org/api/combinereducers#combinereducersreducers) them), contain all the required actions for your app (**type** is the only required field for [action](https://redux.js.org/basics/actions#actions), other fields are up to you). You can create additional folders and files there if you need

**2)** For using Redux with React you will need the [react-redux](https://react-redux.js.org/) library. Don’t forget to wrap the entire components hierarchy Provider component and pass the store as a prop to it.

Each component, which needs access to a redux storecan be wrapped with [**connect(mapStateToProps, mapDispatchToProps)(YourComponent)**](https://react-redux.js.org/api/connect) or you can use hooks (useDispatch, useSelector)

**3)** Use [**redux-thunk**](https://github.com/reduxjs/redux-thunk) **(**[**purpose of redux-thunk**](https://stackoverflow.com/questions/35411423/how-to-dispatch-a-redux-action-with-a-timeout/35415559#35415559)**,** [**how to apply it to your store**](https://redux.js.org/api/compose#example)**)** as middleware in orderto ‘teach’ Redux to recognize special kinds of actions that are in fact functions. This middleware will help you to write the following ‘action-function’:

