1. Why redux?

To be starting with Redux is an open-source Javascript library for managing application state.The main purpose of redux is
it's mostly used as a state management tool ,let's see some benefits of using redux
 . Ease in maintenance
 . Server rendering
 . Great supportive community and so on.....

2.How to install and setup Redux?

To install Redux in your system you need to have Node.js ,once Node.js is installed search for NPM and type the following
(Create-react-app myapp) the word myapp is the file name you can give any name according to your project
 if the word (Happy-Hacking) is displayed it means your react-app is succesfully installed

Now we have to install various dependencies required for our project which is easily availanle in npm store
i.e npm install redux-devtool-extension and so on.....

3.Explain store in Redux?

Before talking about store,let's get to know about the concept called "Flux" flux is an application architecture that used for building client-side web application with react

we can witness the following in flux concept
1.View
2.Action
3.Reducer
4.Root reducers(for combining multiple reducers)
5.store

Now coming back the store consists the application state and logic.it is used for maintaining a particular state within the application

4.How to set Initialstate in Redux?

There are two main ways to initialize state for your application. The createStore method can accept an optional preloadedState value as
 its second argument. Reducers can also specify an initial value by looking for an incoming state argument that is undefined , and returning the value they'd like to use as a default.

5.How to dispatch an Action ?

Firrst we need to import "useDispatch" from React-redux ,let dispatch =useDispatch() if there are 2 buttons you could possibly
use in tge following way

```
    let dispatch =useDispatch()
```

```
   let loginHandler=()=>{
       dispatch(loginAction())

     }
     let logoutHandler =()=>{
         dispatch(logoutAction())

     }
```

6.What is Redux Action ?

Action contains action and action types which will invoke and further dispatched to reducer ,When something happens in the app:
The UI dispatches an action. The store runs the reducers, and the state is updated based on what occurred. The store notifies the
UI that the state has changed.following is example of action and action types

```
const login="login"
const logout="logout"

let loginAction = () =>{
    return{
        type:login
    }

}

let logoutAction = ()=>{
    return{
        type:logout
    }

}

export {login,logout,loginAction,logoutAction}
```

6.What is pure function?Explain Pure function vs Impure function ?

Pure Function is a function (a block of code) that always returns the same result if the same arguments are passed. It does not depend on any state,
or data change during a program's execution rather it only depends on its input arguments

A pure function must both be predictable and without side-effects. ... An impure function is kind of the opposite of a pure one - it doesn't predictably
produce the same result given the same inputs when called multiple times, and may cause side-effects

7.What are Reducers in Redux ?

In Redux, a reducer is a pure function that takes an action and the previous state of the application and returns the new state. The action describes
what happened and it is the reducer's job to return the new state based on that action.

8.How to access redux store outside a react component ?

you need to dispatch actions from outside a React component, the same technique will work: import the store, then call store. dispatch() passing the action you need to dispatch. It works the same as the dispatch function you get from props via react-redux's connect function.

9.How to use connect from React-Redux ?

The connect() function connects a React component to a Redux store.

It provides its connected component with the pieces of the data it needs from the store, and the functions it can use to dispatch actions to the store.

It does not modify the component class passed to it; instead, it returns a new, connected component class that wraps the component you passed in.

10.How to add multiple middleware's to Redux ?

This is how to apply one or many middlewares :

```
import {createStore, applyMiddleware} from 'redux';
import thunk from 'redux-thunk';
import logger from 'redux-logger';
import {rootReducer} from "../reducers"; ( Import your combined reducers)

const middleWares = [thunk, logger]; (Put the list of third part plugins
in an array)


export const store = createStore(rootReducer,
applyMiddleware(...middleWares));
```

Now import the store exported recently in your index.js and pass it to the Provider component.

To view the example code of Flux concept that is View,Action,Reducer,Root Reducer,Store you can follow the below link

https://github.com/playerunknownL/mywork/tree/main/src/Practice