Redux with Hooks and Functional Components

1. Create a React.js application using React CLI
2. Install following. Packages
   a. npm install - -save **redux react-redux**
3. In the application add the following folders
   a. action
   b. reducers
   c. components
4. In actions folder add the following code by adding actions.js file in it

```
const addProduct=(product)=>{



    return{

        type: "ADD_PRODUCT",

        product

    }

}



export default addProduct;
```

In the reducers folder add the following code by adding reducers.js file in it

```
import addProduct from './../actions/actions';

import {combineReducers} from 'redux';

export const addProductReducer=(state, action)=>{



    switch(action.type){
```

```
        case 'ADD_PRODUCT':



            return {


                product:action.product


            };


        default:


                return state;


    }


};




export const listProductReducers=(state=[],action)=>{




    switch(action.type){


        case 'ADD_PRODUCT':




                return [...state, addProductReducer(undefined,action)];


        default:


                return state;


    }
```

```
};



const rootReducer = combineReducers({listProductReducers});

export default rootReducer;
```

In the components folder add a new file and name it as AddPorductComoponent.jsx and add the following code in it

```jsx
import React, { useState } from 'react';




const AddProductCononent=(props)=>{

    const [product, updateProduct] =

useState({ProductId:0,ProductName:''});

    const save=()=>{

        props.AddClick(product);

    }

    const clear=()=>{



        updateProduct({ProductId:0,ProductName:''});

    }
```

```jsx
    return (

        <div className="container">

            <div className="form-group">

                <label>Product Id</label>

                <input type="text" className="form-control"

                value={product.ProductId}

                 onChange={(evt)=> {updateProduct({...product, ProductId:

parseInt(evt.target.value)})}}/>

            </div>

            <div className="form-group">

                <label>Product Name</label>

                <input type="text" className="form-control"

                value={product.ProductName}

                onChange={(evt)=> {updateProduct({...product, ProductName:

evt.target.value})}}

                />

            </div>

            <div className="form-group">

            <input type="button" value="Clear"

            className="btn btn-warning"
```

```jsx
            onClick={clear}/>

                <input type="button" value="Add Product"

                className="btn btn-success"

                onClick={save}/>

            </div>

        </div>

    );

}


export default AddProductCononent;
```

Add a new file in components folder and name it as ListProductsComponent.jsx. Add the following code in this file

```jsx
import React, { useState } from 'react';




const AddProductCononent=(props)=>{

    const [product, updateProduct] =

useState({ProductId:0,ProductName:''});

    const save=()=>{

        props.AddClick(product);

    }
```

```
    const clear=()=>{



        updateProduct({ProductId:0,ProductName:''});

    }

    return (

        <div className="container">

            <div className="form-group">

                <label>Product Id</label>

                <input type="text" className="form-control"

                value={product.ProductId}

                 onChange={(evt)=> {updateProduct({...product, ProductId:

parseInt(evt.target.value)})}}/>

            </div>

            <div className="form-group">

                <label>Product Name</label>

                <input type="text" className="form-control"

                value={product.ProductName}

                onChange={(evt)=> {updateProduct({...product, ProductName:

evt.target.value})}}

                />
```

```jsx
        </div>

        <div className="form-group">

        <input type="button" value="Clear"

        className="btn btn-warning"

        onClick={clear}/>

            <input type="button" value="Add Product"

            className="btn btn-success"

            onClick={save}/>

        </div>

    </div>

    );

}




export default AddProductCononent;
```

Modify App.js with following code

```jsx
import React from 'react';

import {useSelector, useDispatch} from 'react-redux'

import addProduct from './actions/actions.js';



import AddProductComponent from './components/AddProductComponent.jsx';
```

```jsx
import ListProductsComponent from

'./components/ListProductsComponent.jsx';


const App=()=> {

    let products = useSelector(state=>state.listProductReducers);


    let dispatch = useDispatch();


  return (

    <div className="container">

        <AddProductComponent

AddClick={(product=>dispatch(addProduct(product)))}></AddProductComponent

>

        <hr/>

        <ListProductsComponent

listProducts={products}></ListProductsComponent>

    </div>

  );

}
```

```
export default App;
```

Add the following code in index.js

```
import React from 'react';

import ReactDOM from 'react-dom';

import {createStore, applyMiddleware, compose} from 'redux';

import {Provider} from 'react-redux';

import rootReducer from './reducers/reducers';

import './index.css';

import './../node_modules/bootstrap/dist/css/bootstrap.min.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

  const store = createStore(reducerdata,
```

```
    window.__REDUX_DEVTOOLS_EXTENSION__  &&

window.__REDUX_DEVTOOLS_EXTENSION__() );

    ReactDOM.render(

      <Provider store={store}>

      <App />

     </Provider>,

     document.getElementById('root')

    );


// If you want to start measuring performance in your app, pass a

function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-

vitals

reportWebVitals();
```

Run the application using the **npm run start** command. Add the new product info and click on save button. The List products component will display theninormation.