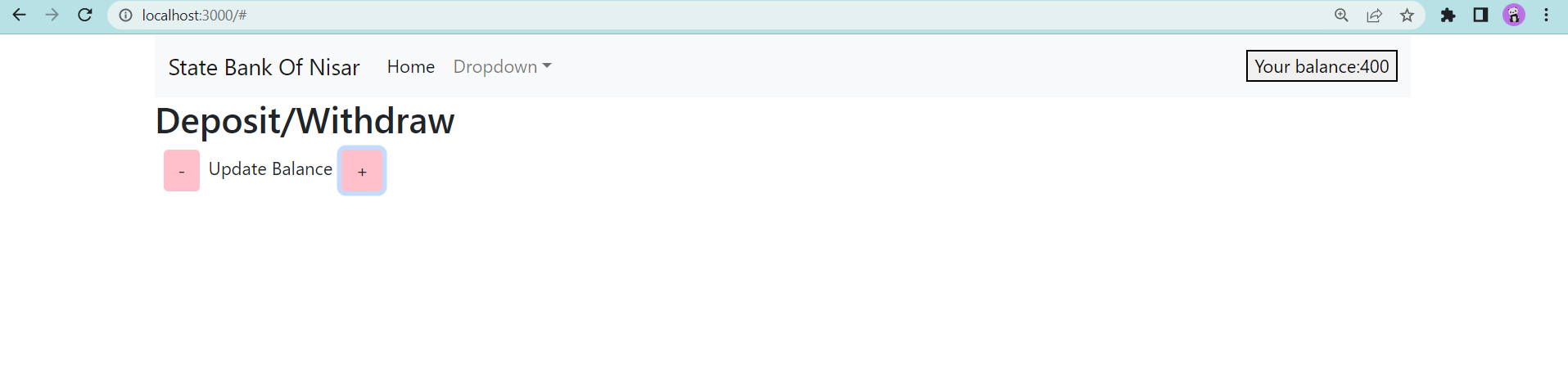
Redux Notes

Step 1:- Create React App…REDUXAPP.

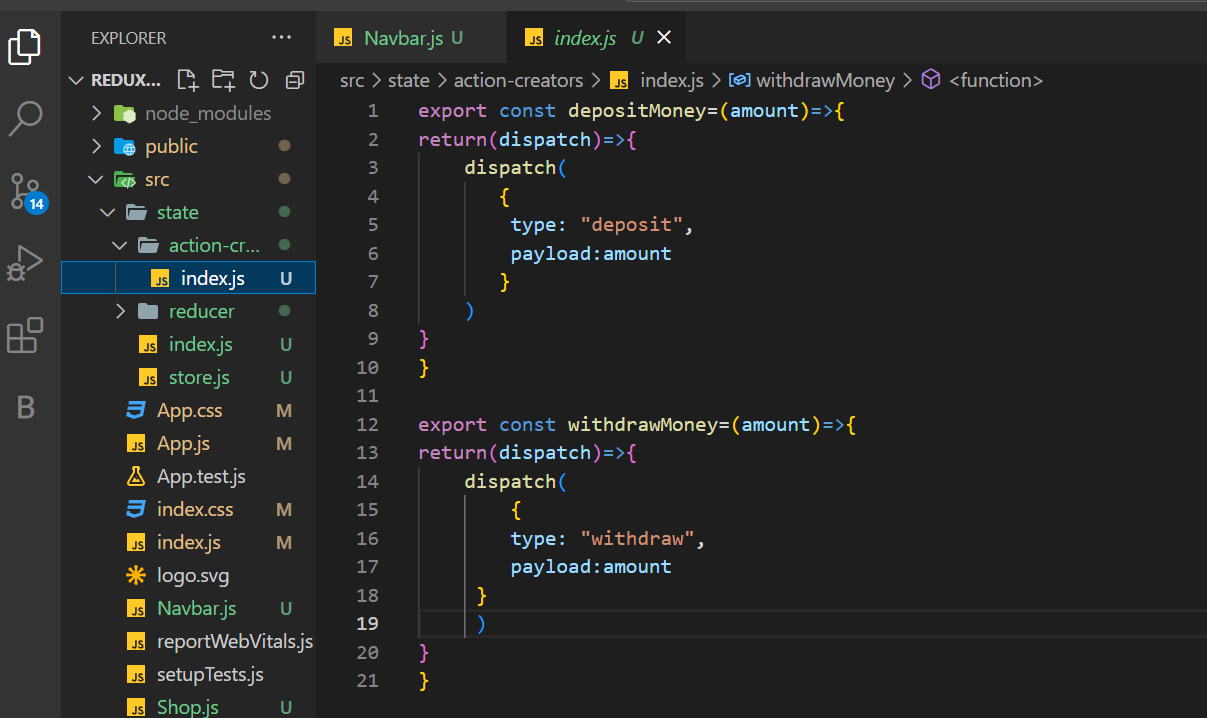
Step 2:- Import Bootstrap For Better Visualisation.

{ A] Create a component NAVBAR.JS containing simple navbar and Your Balance button.

B].Create a Component SHOP.JS Containing Deposit/Withdraw text on UI}

Step 3:- Create A Store Folder.

Step 4 :- Make Action-Creators Folder.

{ A] Create a Dispatch Function Containing Type And Payload.}

Step 5 :- Make Reducer Folder.

Step 6 :- Make a AmountReducer.js File In Reducer Folder.

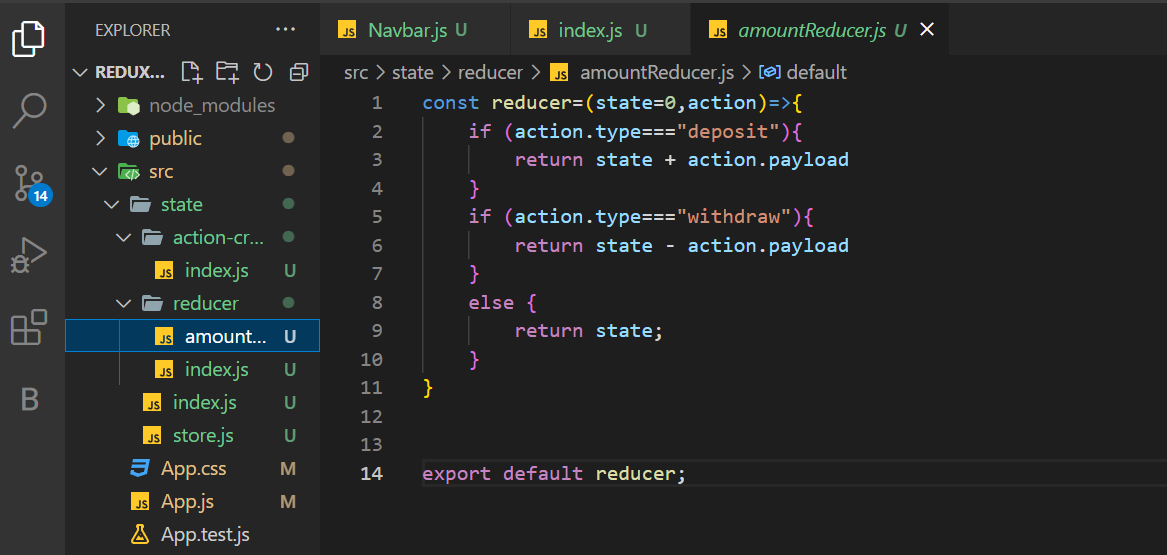
{

A].Create a Reducer Function. It Will take two argiments :

a]Previous State b] Action

B]After Performing action it will return New State.

}



Step 6 :- Make a Index.js File In Reducer Folder.

{ A]Create a Reducers=combineReducers Function.

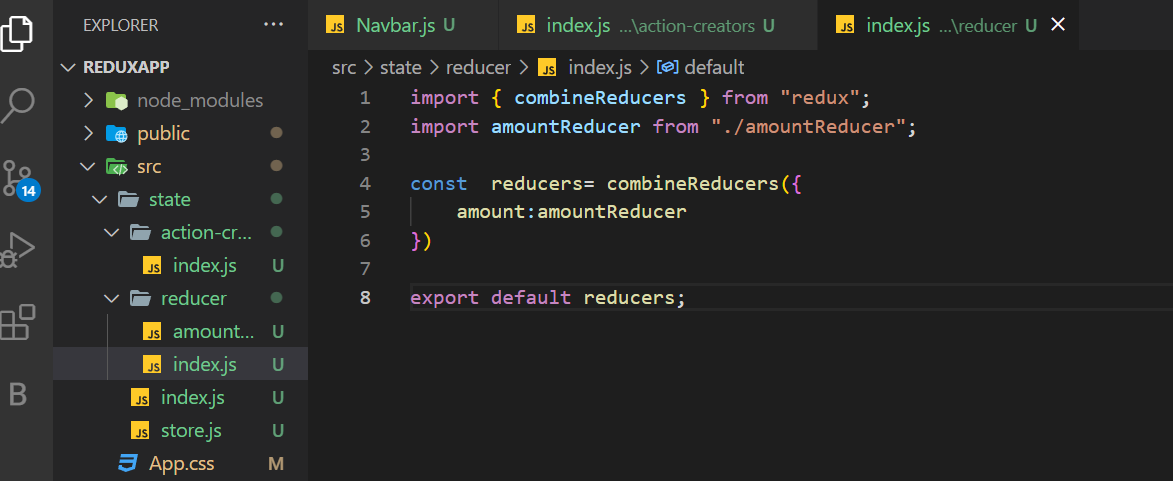
It will Combine all the reducers present in the app.

B]Import All Reducers in CombineReducer Such As..

amount:amountReducer,// For Single Reducer

SecondReducer:SecondReducer // For Multiple Reducer

}



\*Step 7 :- Make a Index.js File In State Folder.

{Export \* as actionCreators ..}

export \* as actionCreators from "./action-creators/index"

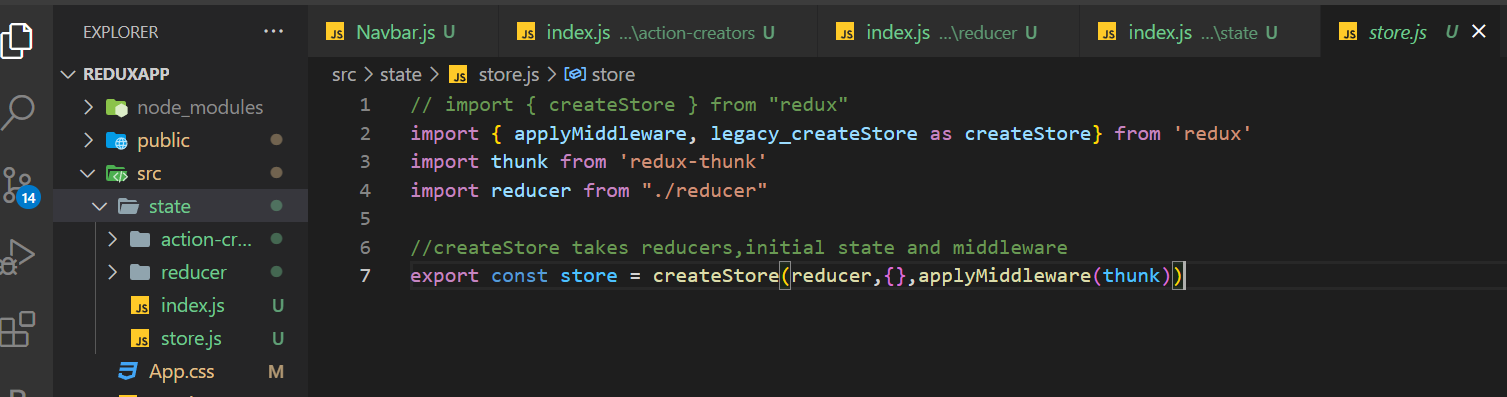
Step 8:- Make a Store.js File In State Folder.

{Make a Function Store= createStore(reducer,{},middleware(thunk)..

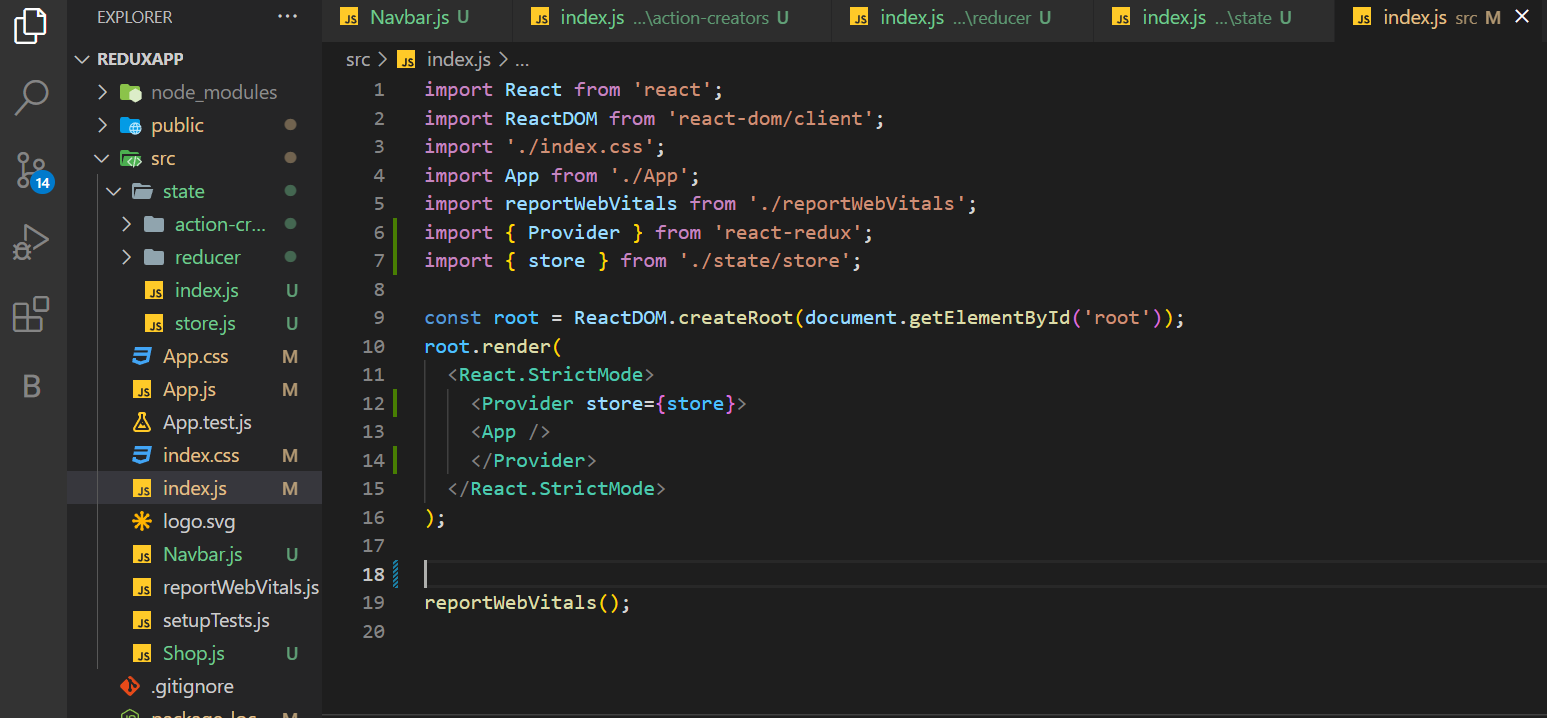
// createStore will take 3 arguments

a] reducer b] initial state c] middleware

}

  
  
  
Step 9:- Wrap Your App inside Provider.

{Wrap your main App inside provider such as below Picture..

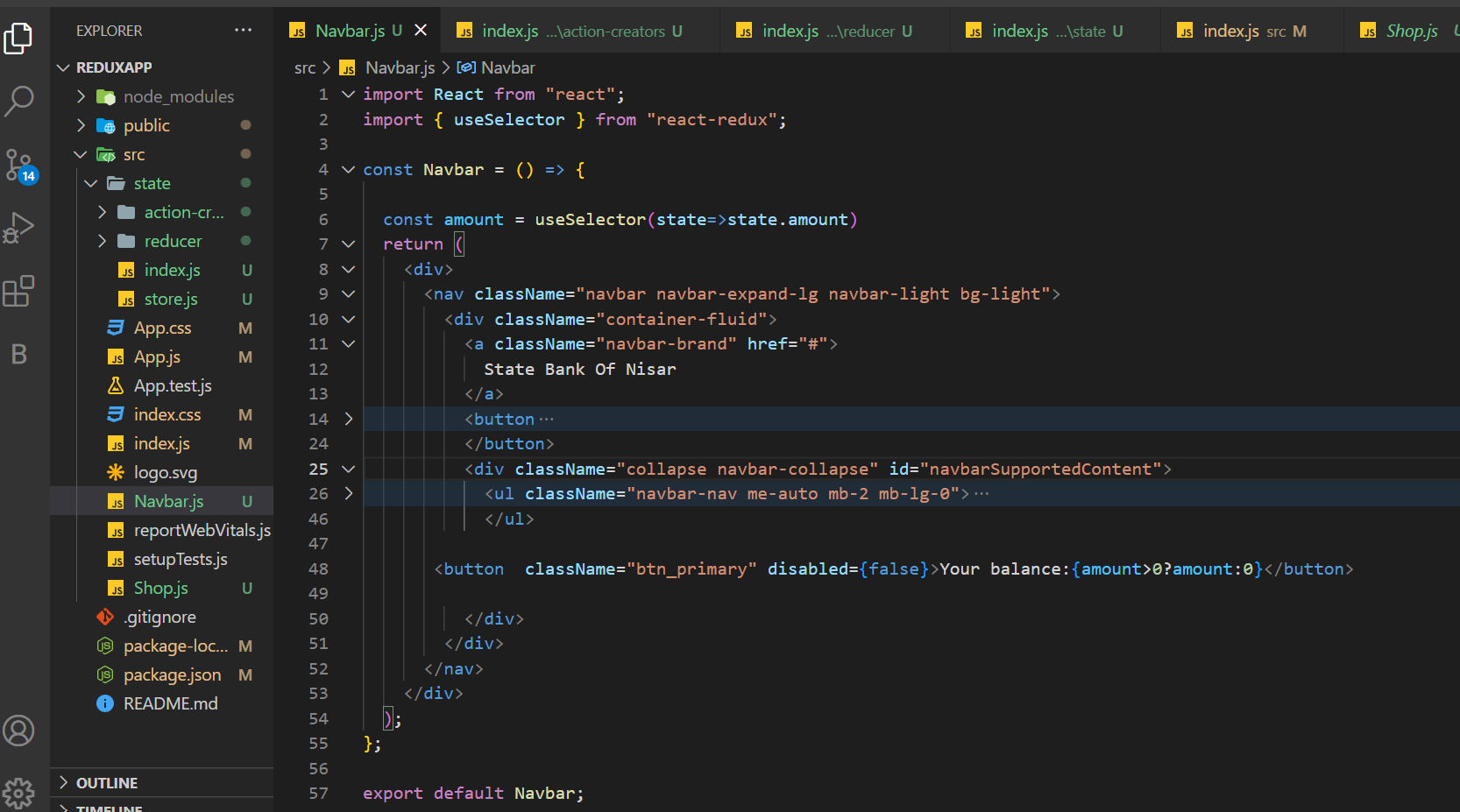
//It will give App acess to your Store, which will ultimately store your all App data. }

Step 9:- Import UseSelector().

{A] Import USESELECTOR in your Navbar.js so that It waill get acess to the Action{i.e. type,payload}.

B] Implement the Functionality on Your Balance Button.

//line no.6 & 48.  
C] UseSelector is Equivalent to MapStateToProps  
//it takes in a function argument that returns the part of state that you want..}



Step 10:- Import UseDispatch().

{A]. Import USEDISPATCH() from redux, in SHOP.JS.

We can invoke UseDispatch and store it to a variable …dispatch.

Dispatch will work with the all Actions imported from action folder  
//line no.8….assinging to variable

//line 15 &23 …adding functionality

B]. USEDISPATCH is Equivalent to MapDispatchToProps  
// We can also use BindActionCreator() to bind all the actionCreators..}

