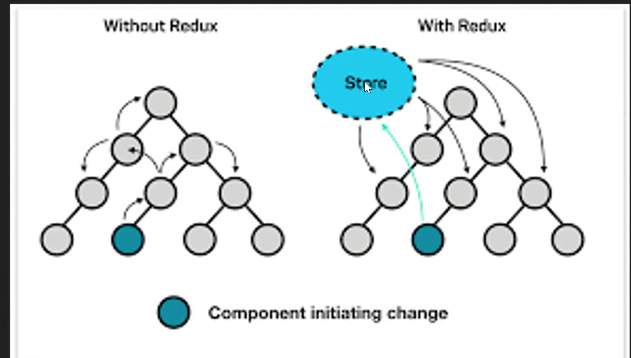


Redux là thư viện của js

Muốn state mà mọi componet sử dụng được => redux

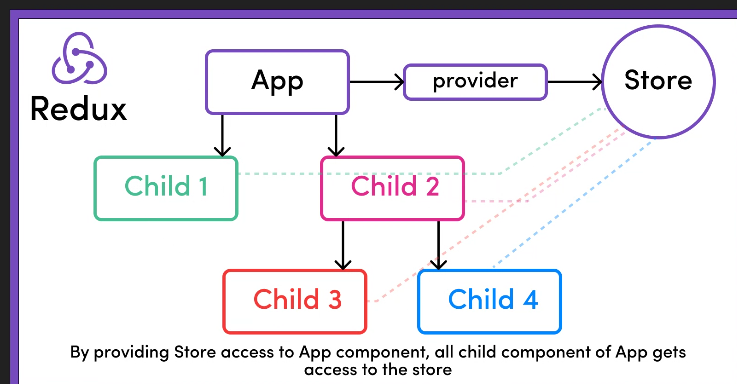
Truyền component khác qh cha con nó sẽ phải truyền qua nhiều component trung gian =>lãng phí

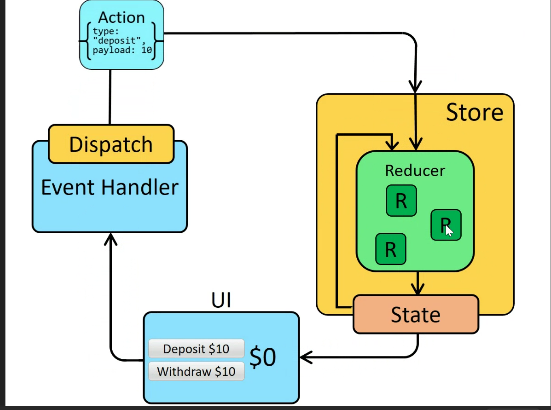
* Sử dụng redux



* Dạng như làm 1 bộ biến toàn cục

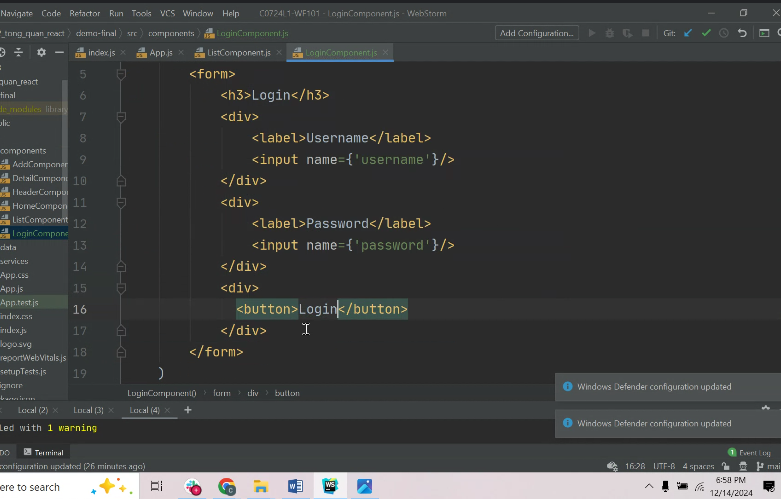
Tránh prop driling là tránh đào sâu drop => nhiều drop đi

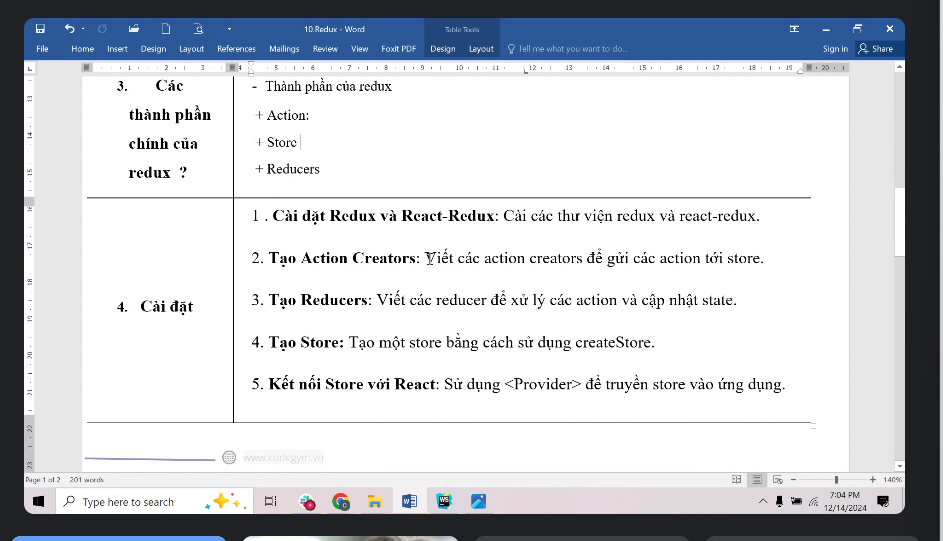


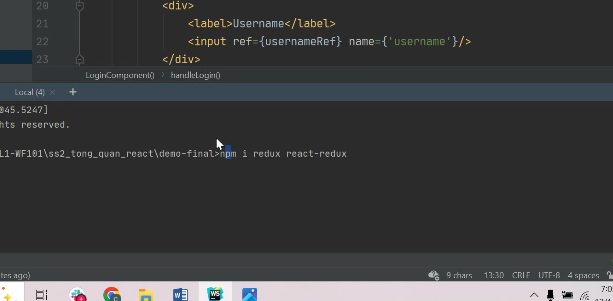


Reducer : quản lý nhiều state

Tạo 1 form riêng:



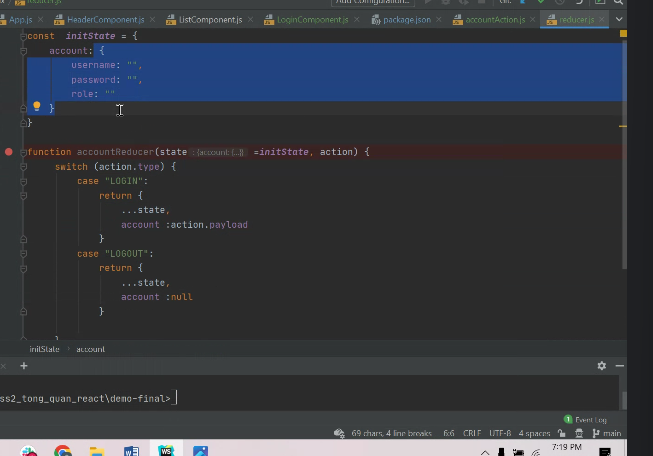


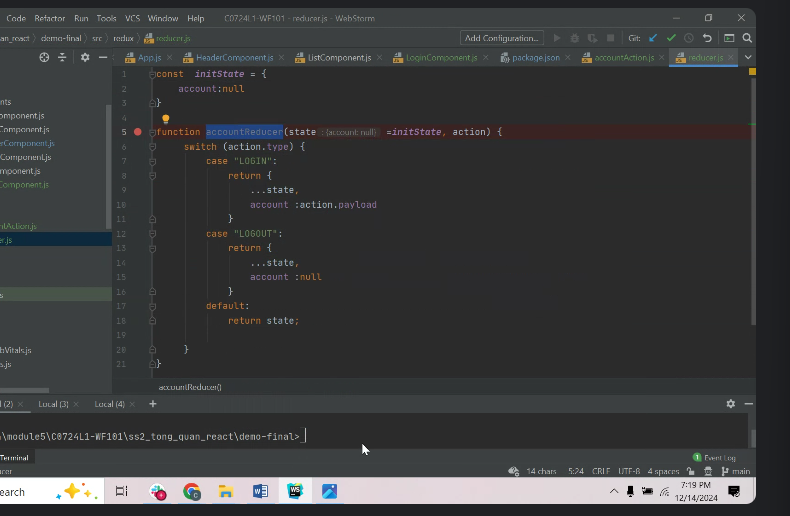


Cài đặt

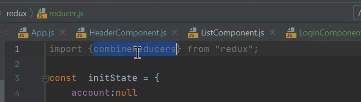
Về reducer nhận 2 tham biến là state, action

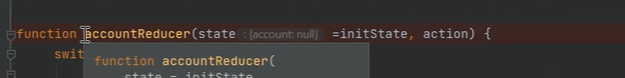
Sử dụng switch case





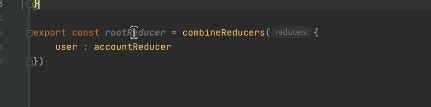
* Đây là reducer gốc(root)



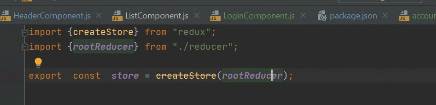


Chứa biến là state là trạng thái state hiên tại, với action là lưu dữ liệu

* Như kiểu gán lại user = cái hàm trên

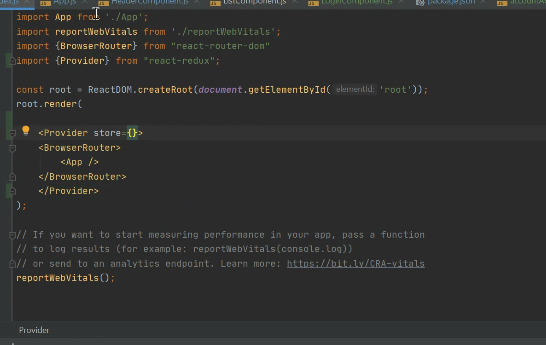


Tạo store:

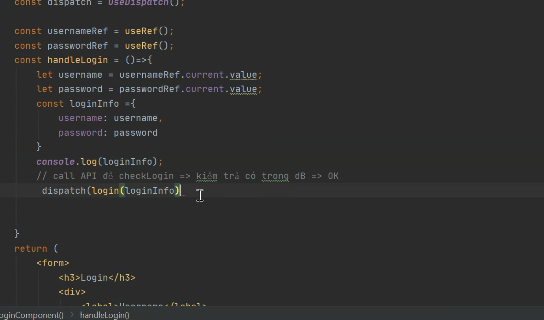


B4: kết nối redux với react

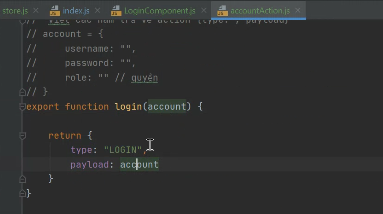
* Lấy thẻ provider để bọc store(props) và bọc cụm cần render



Để nhận action từ nút qua store => sử dụng dispatch

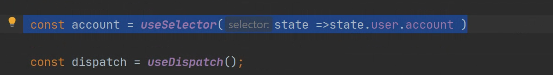


Truyền loginInfor vào

dựa trên <=

Sử dụng hook của dispatch : chuyển lên

useSelector: lấy về



Sử dụng cú pháp && để hiển thị

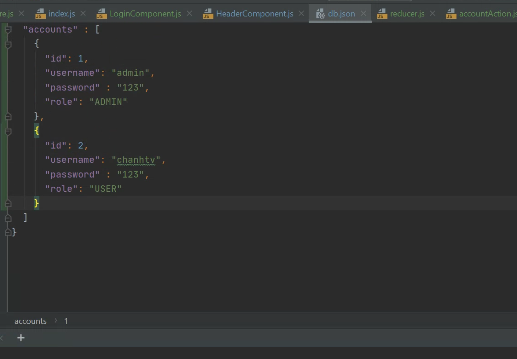


Nav link => chuyển sang 1 component

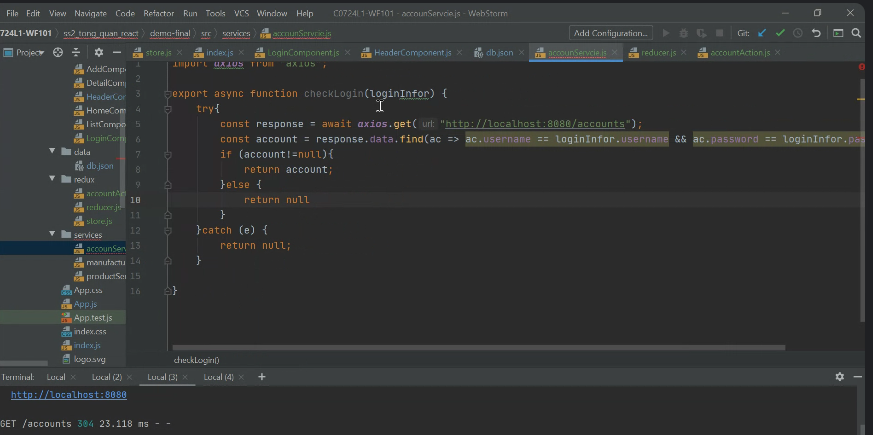
Login logout

Code api:

(mô phỏng backend)



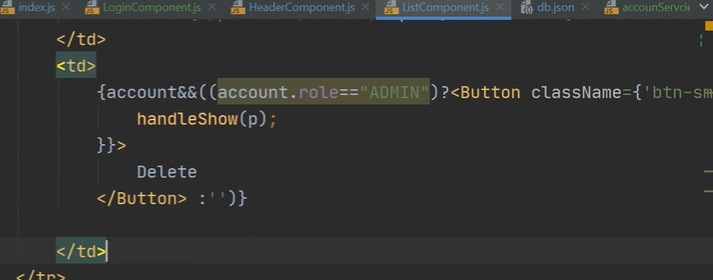
Tạo kho tài khoản



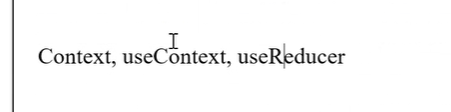
Check tk mk của account: nếu đúng thì trả về đối tượng

Phân quyền admin mới dc xóa ( k có quyền => ẩn nút)

* Sang chỗ hiển thị button thêm điều kiện



Account&& để xét xem account có null hay k => và account.role để điều kiện tên người phân quyền



Tìm hiểu thêm