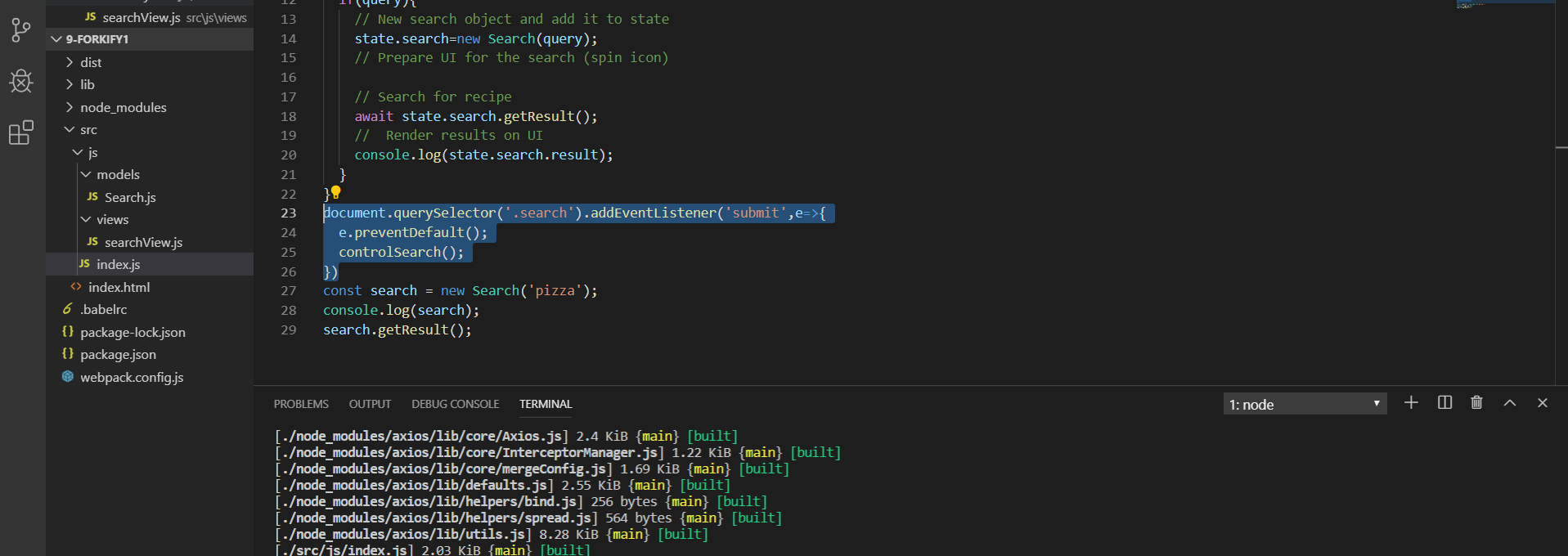


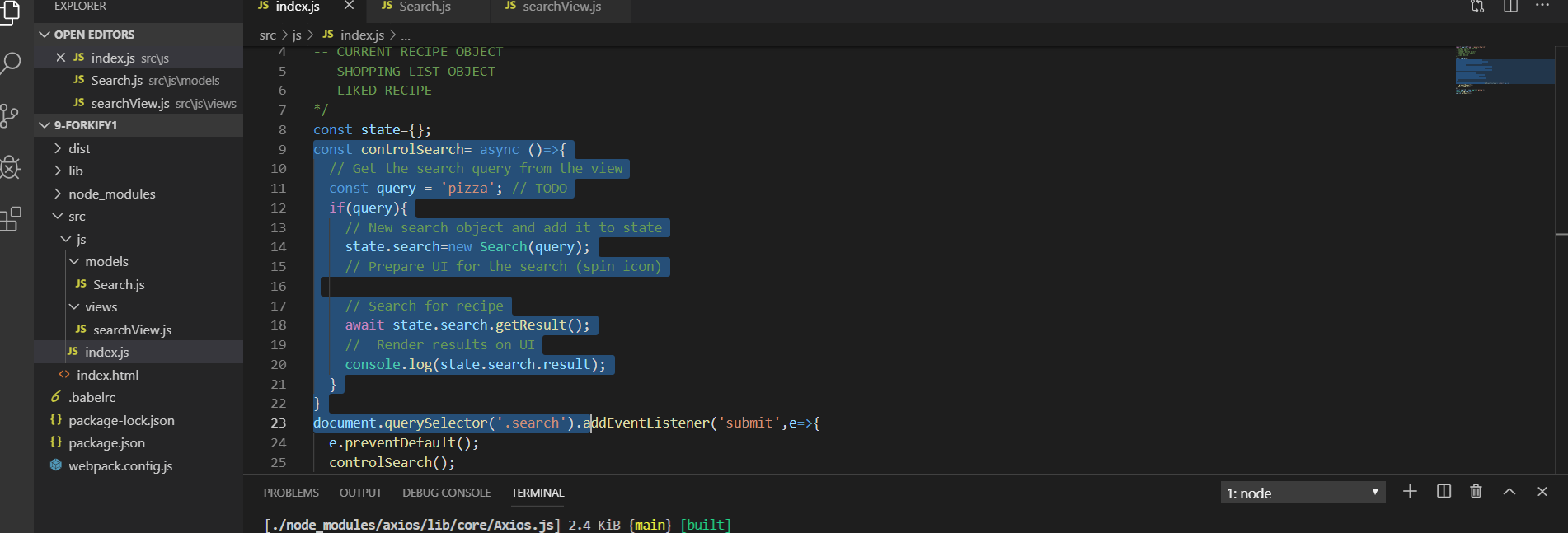
State lưu trạng thái của trang tại mỗi thời điểm, mỗi thời điểm sẽ có những giá trị trong state khác nhau

Khi bắt đầu vào web, state sẽ có một số dữ liệu, và đó là dữ liệu ban đầu cho page, ví dụ các recipe đã like, thì khi load web ta sẽ thấy được những recipe đã like

Việc hiển thị thông qua state thì khi thay đổi state sẽ thay đổi hiển thị

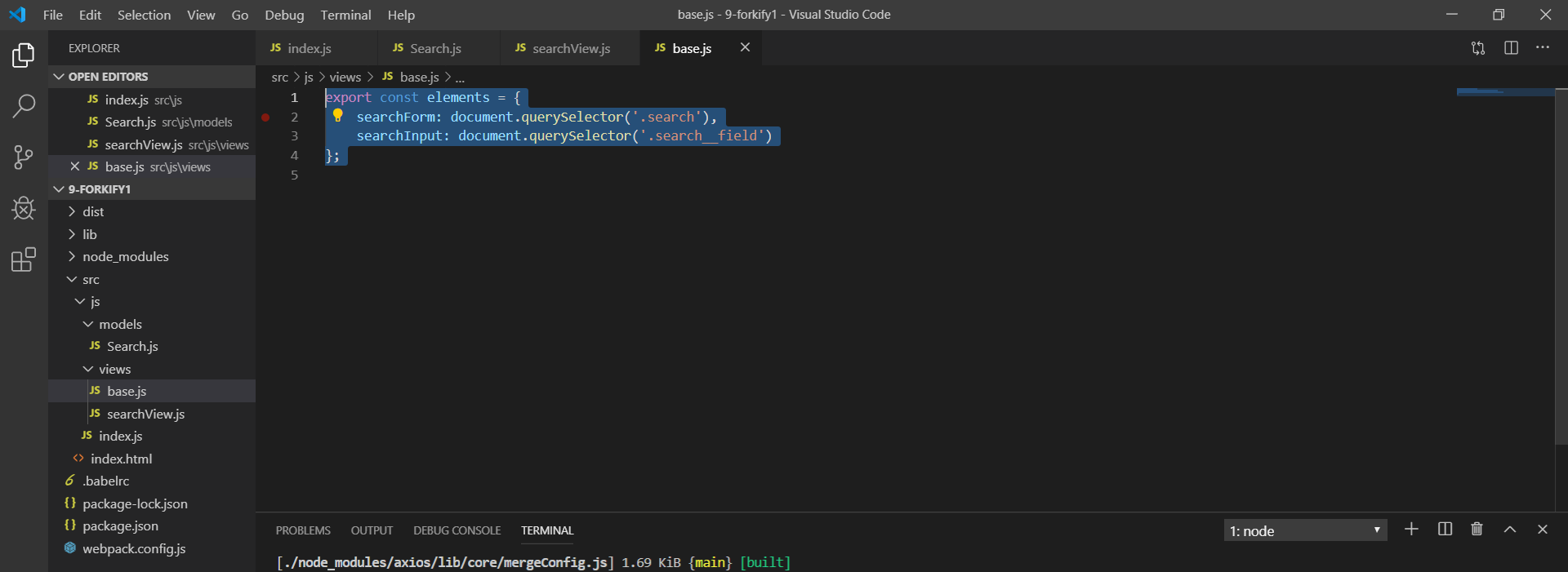


Add event khi submit form, .search là class của form, gọi hàm những việc cần làm sau khi submit vào hàm call back

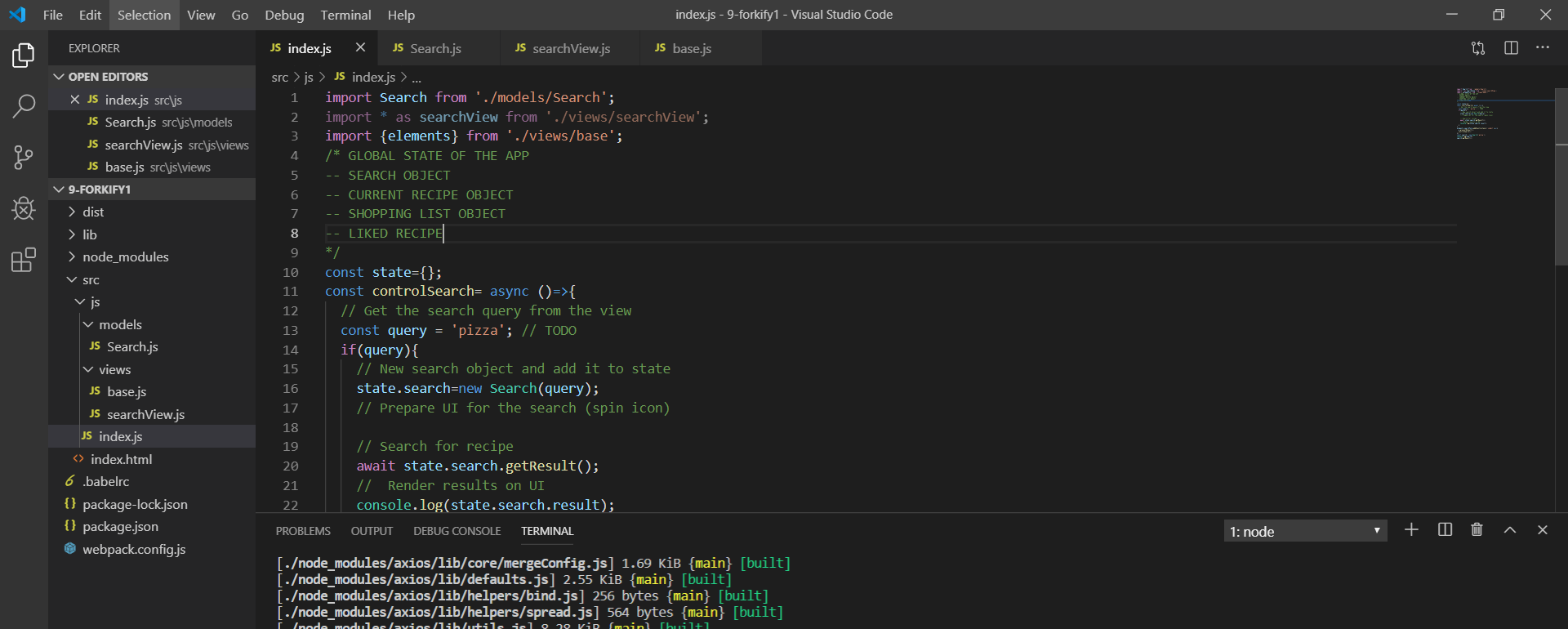


Định nghĩa những việc cần làm

Việc getResult sẽ mất thời gian -> await -> đưa hàm về async

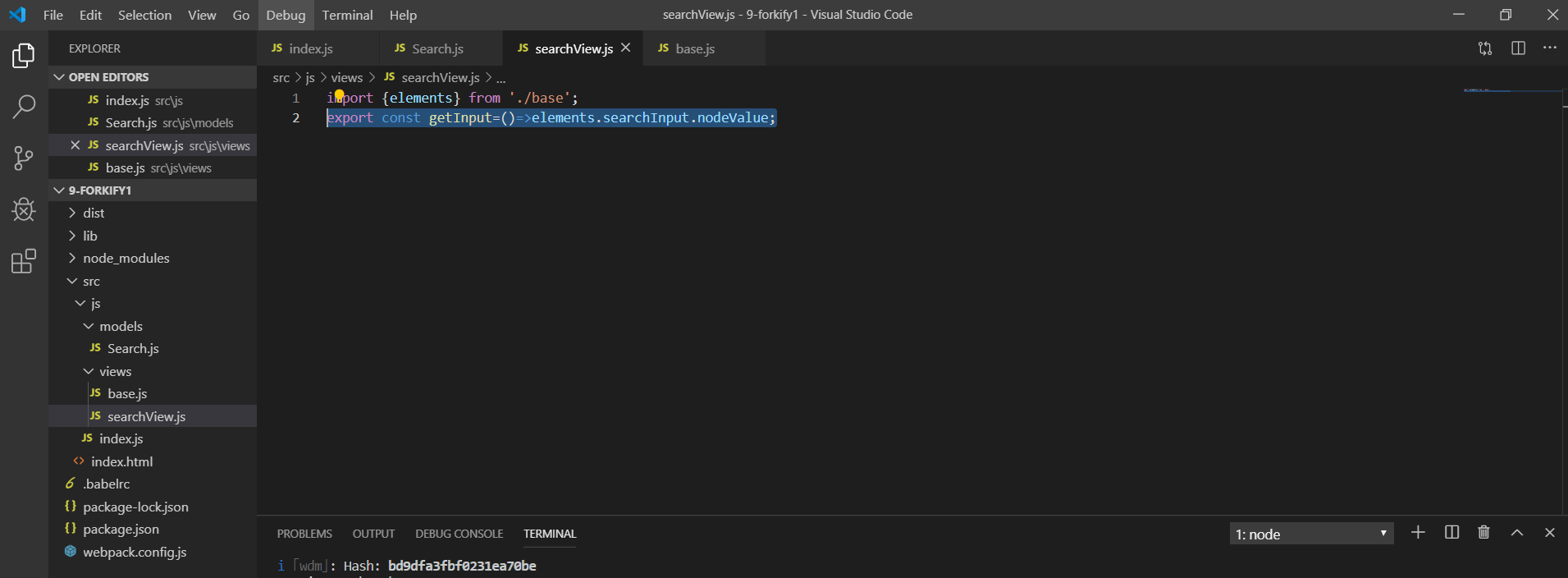


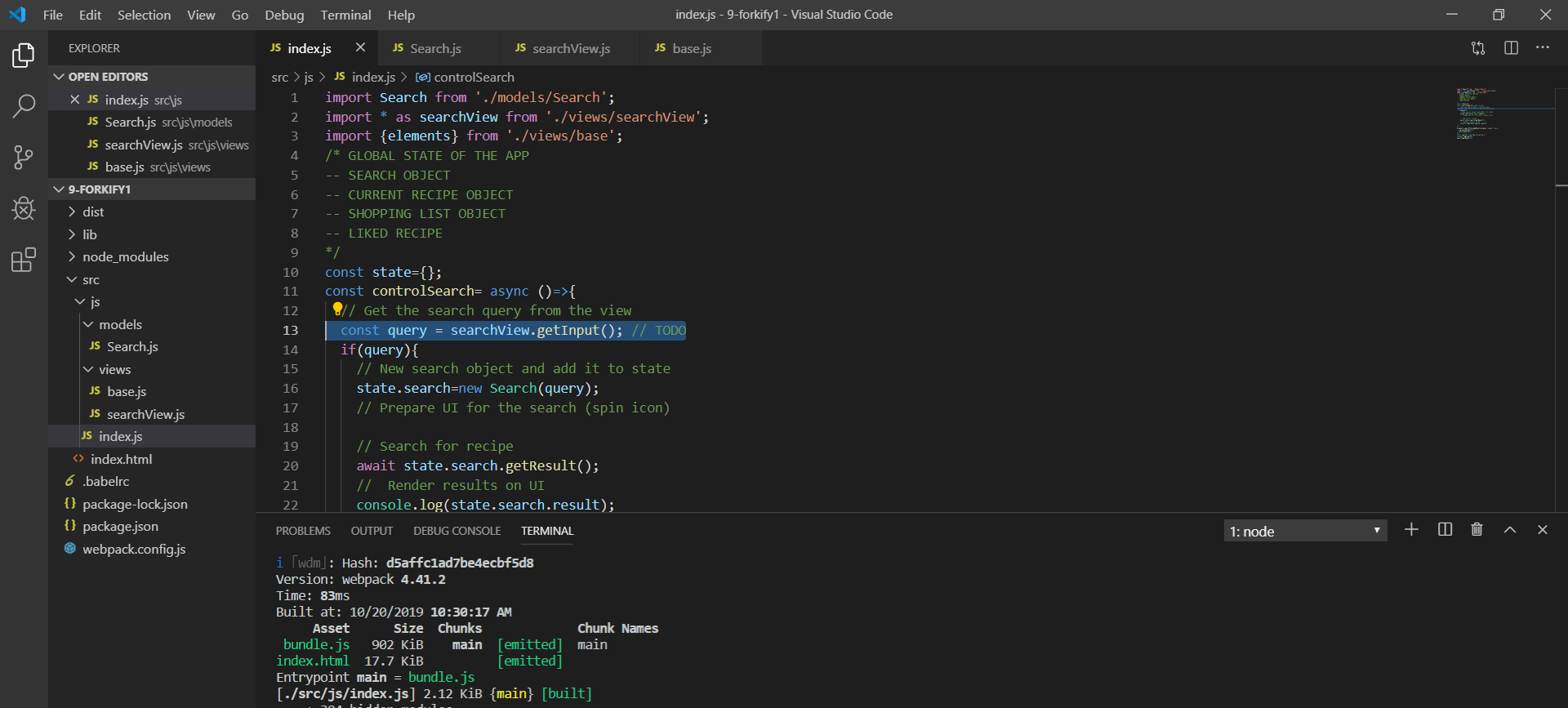
Tạo base.js chứa các selector trong views



Import và sử dụng ở chổ nào cần

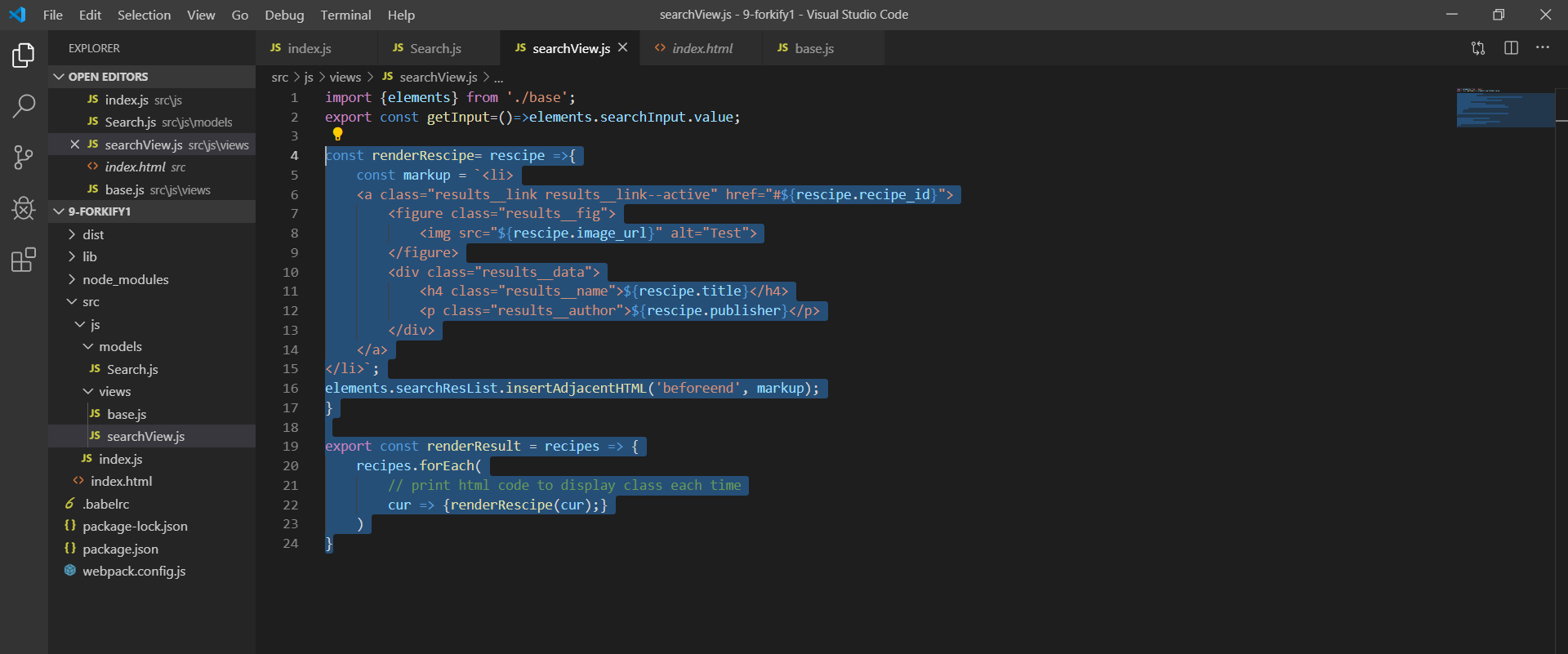
// get value from input

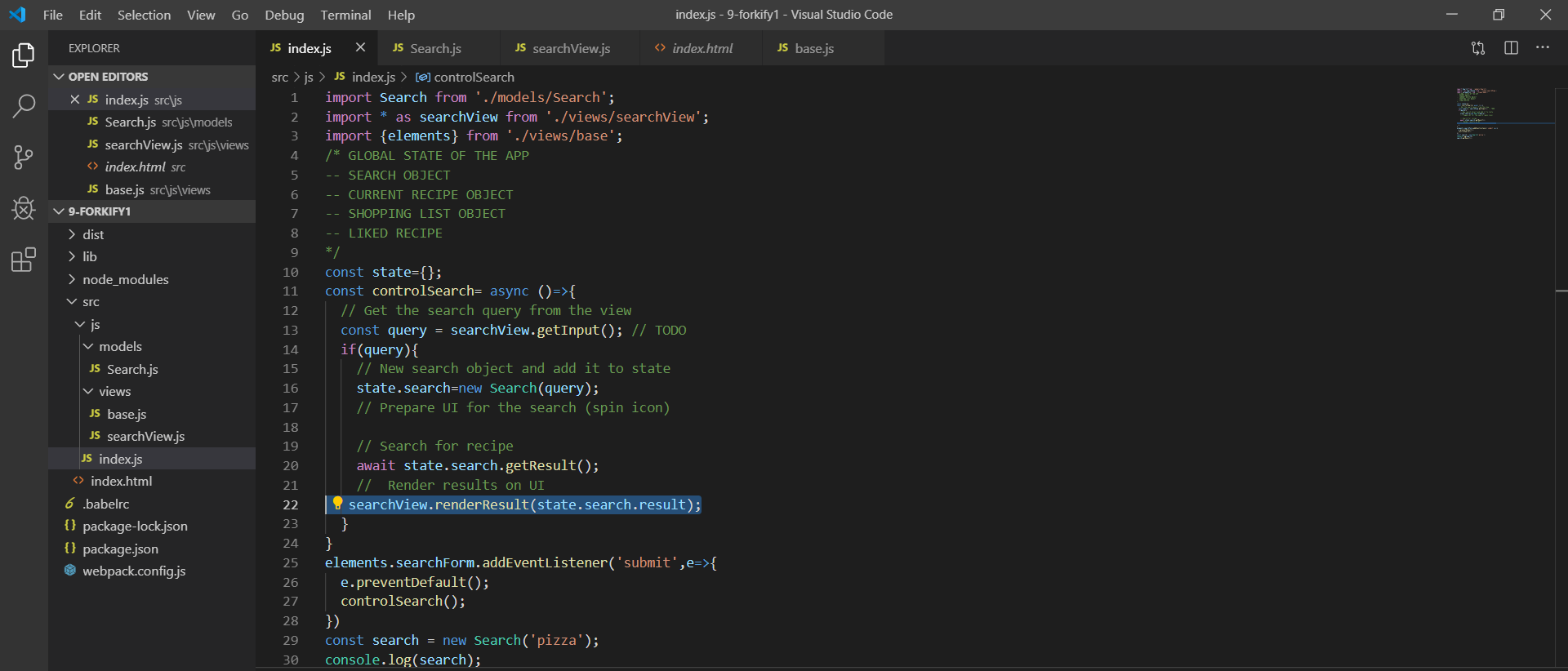




Hàm getinput từ searchview, dùng hàm này ở index.js

// hiển thị kết quả lấy được ra UI





Sử dụng

*element*.insertAdjacentHTML(*position*, *text*);

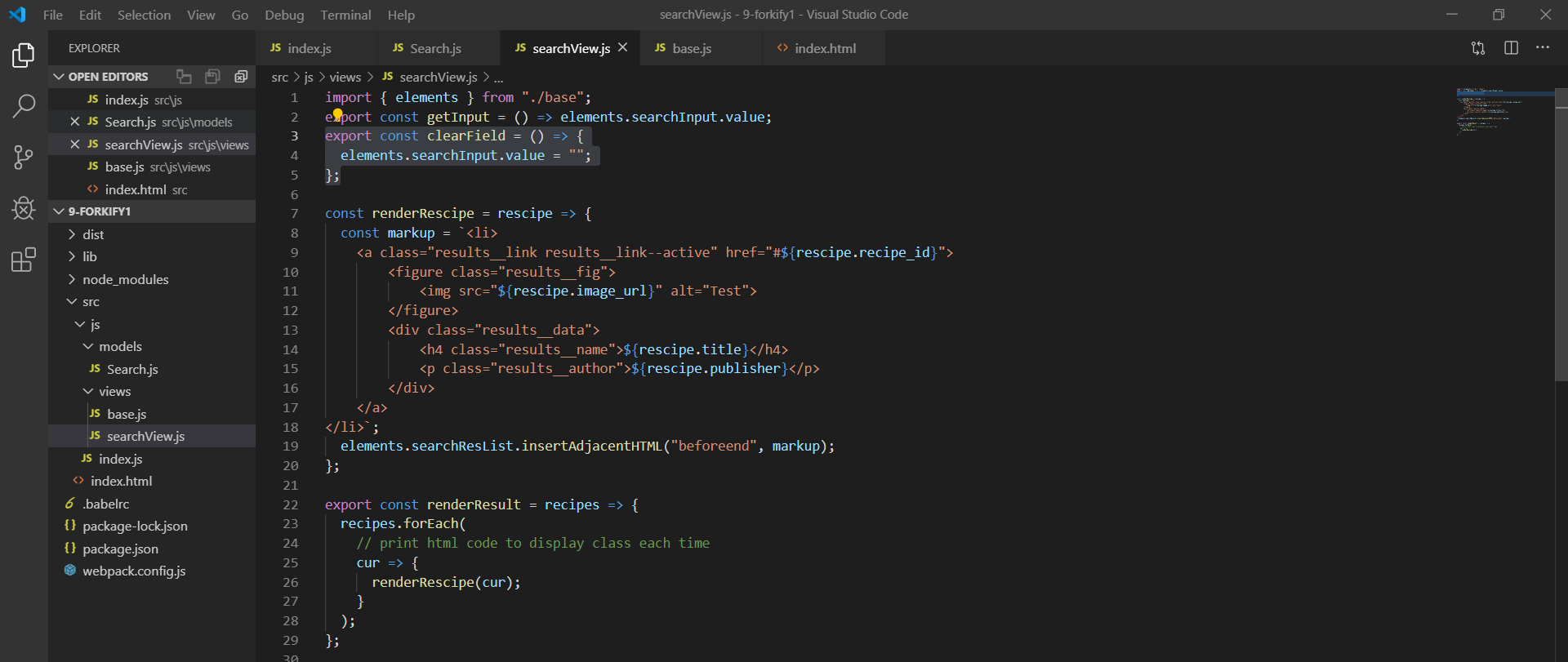
để đưa kết quả hiển thị ra

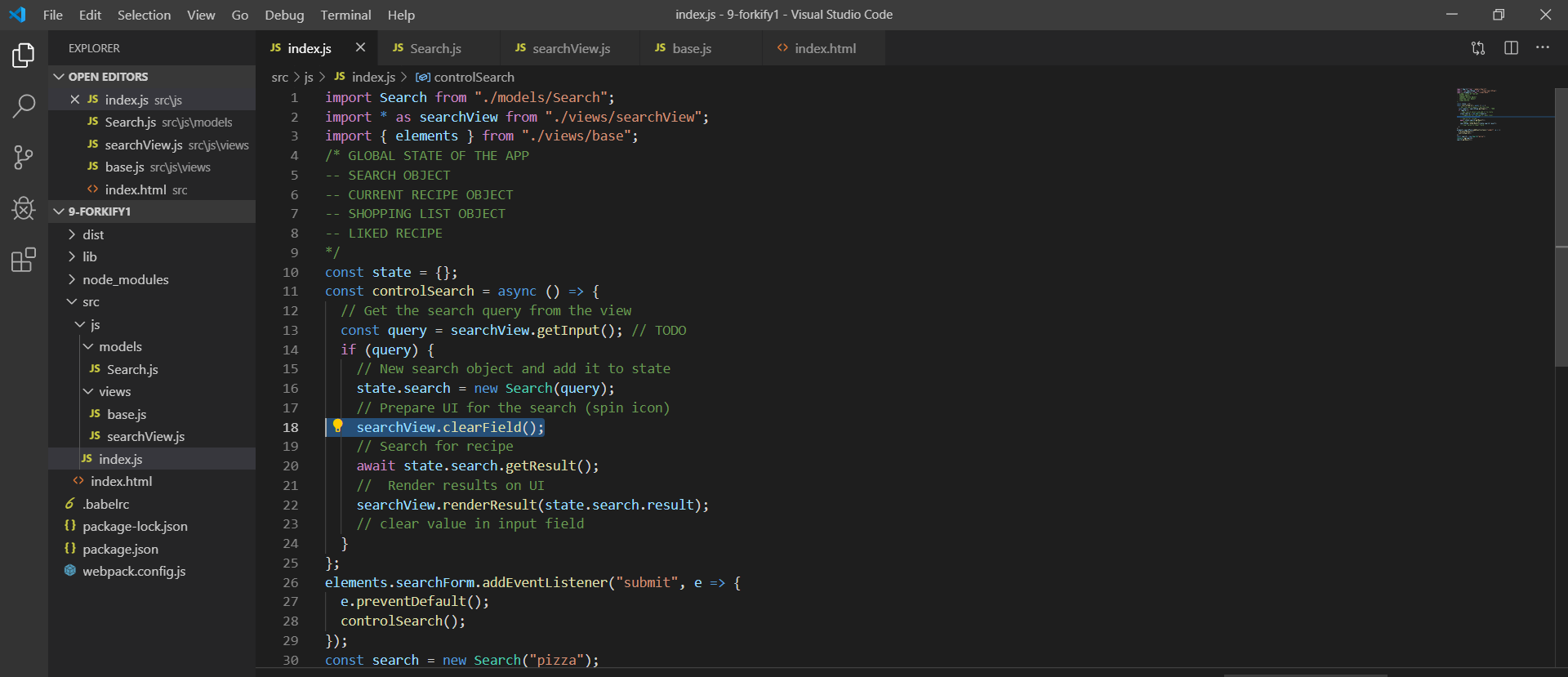
hàm hiển thị dược khai báo trên searchView và được sử dụng ở index.js which is controler

không cần dùng string.replace vì trên es6 đã có ` `

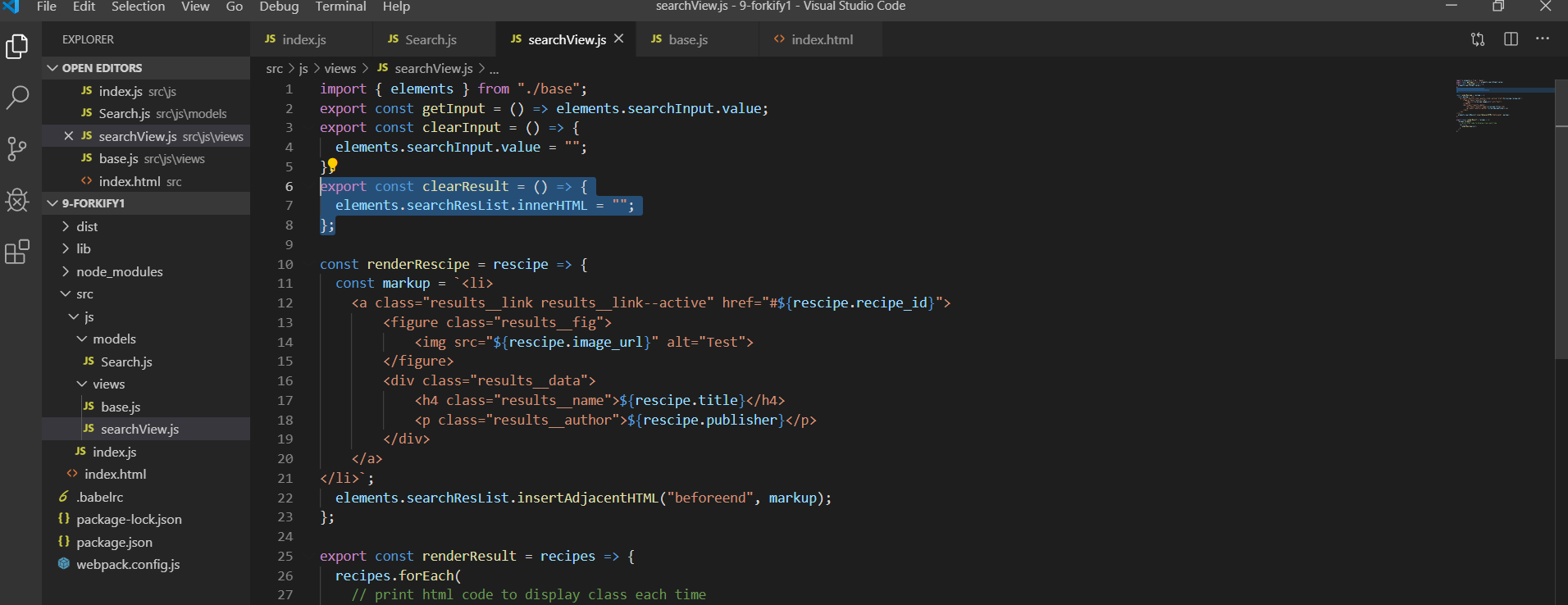
// clear field

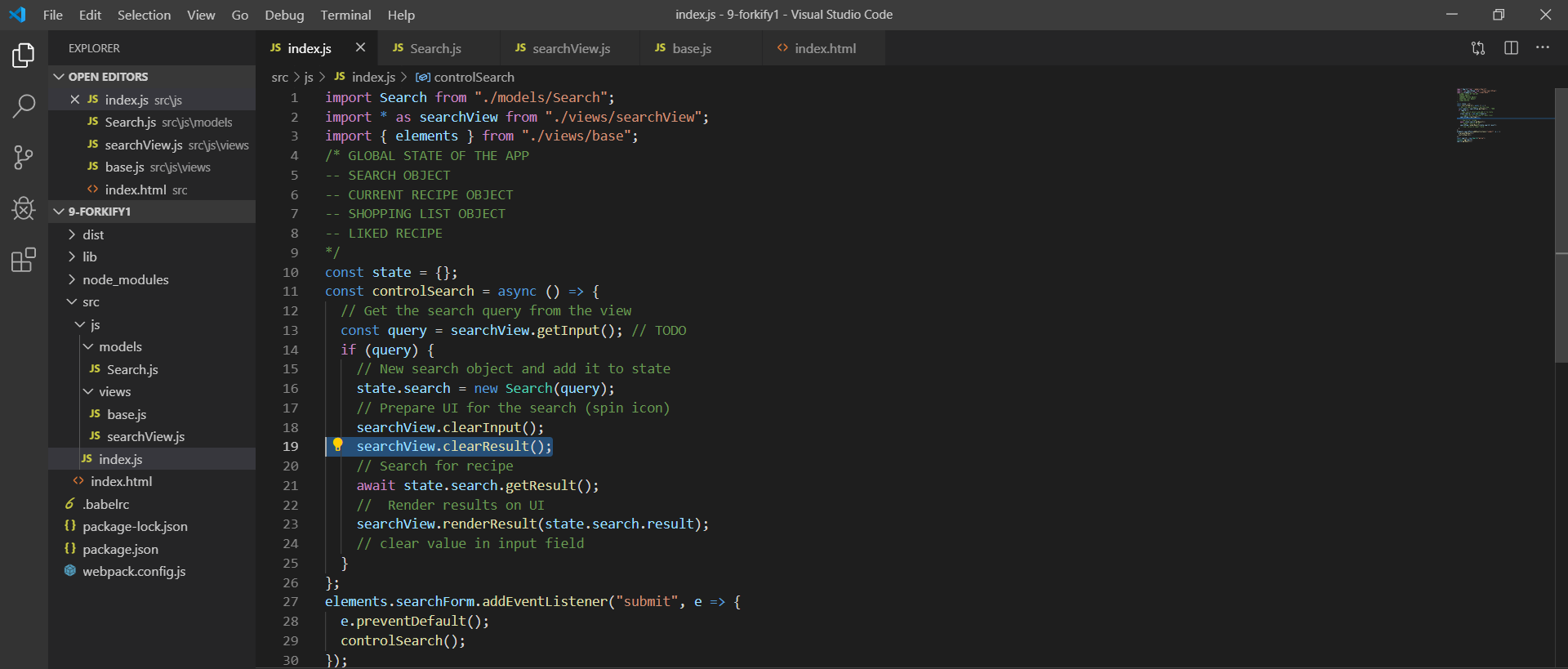
.clear ô input

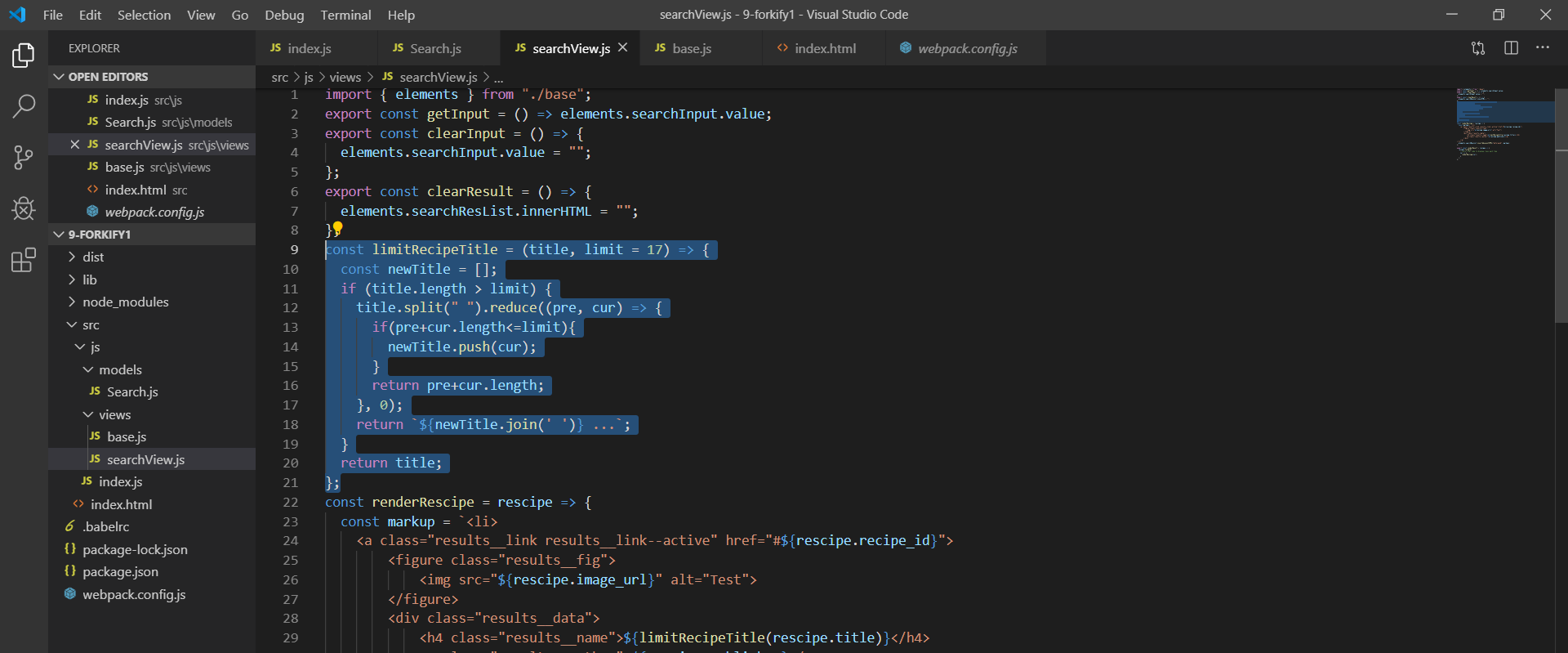




.clear ô hiển thị, vị nếu không clear thì kết quả cũ sẽ còn và kết quả mới sẽ hiện ở dưới



e.preventDefault(); kick button page sẽ reload -> loại bỏ điều đó



Format title trên một dòng

Đặt giới hạn hiển thị, nếu dưới dưới dới hạn đó thì return, còn trên giới hạn đó

Cắt chuỗi ra, nếu độ dài chỗi + từ tiếp theo nhỏ hơn giới hạn thì đưa từ mới vào mangx

Ra khỏi vòng lặp thì dùng array.join để ghép mãng lại