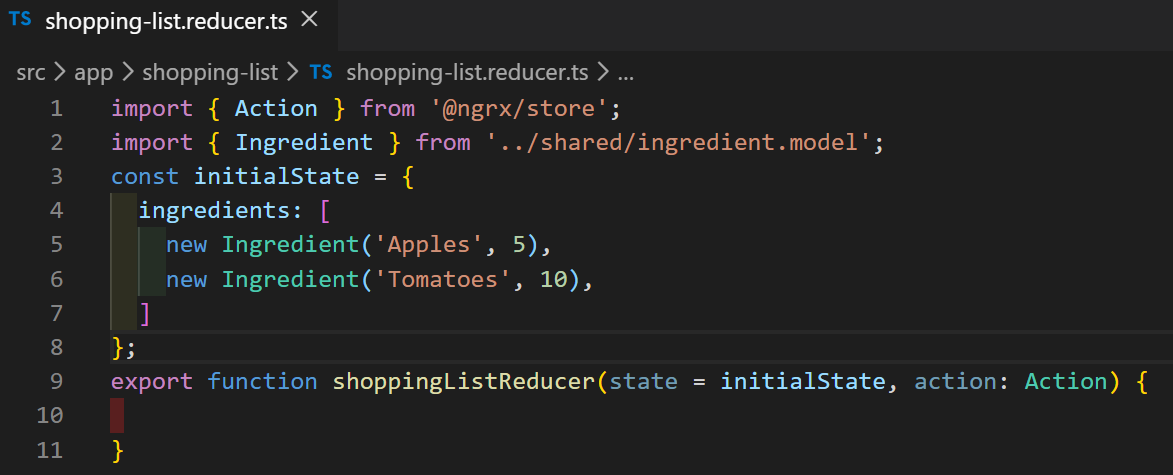
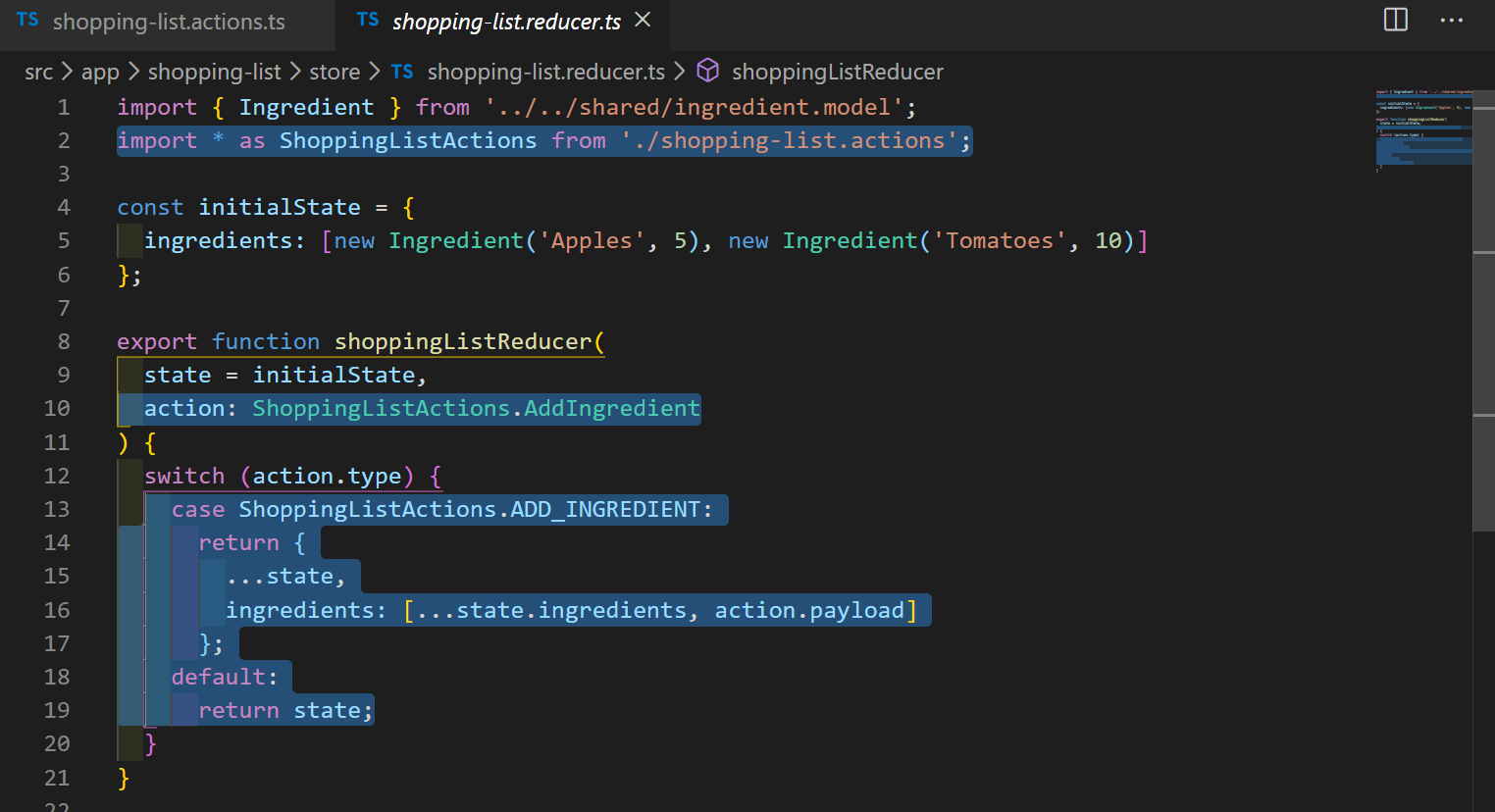
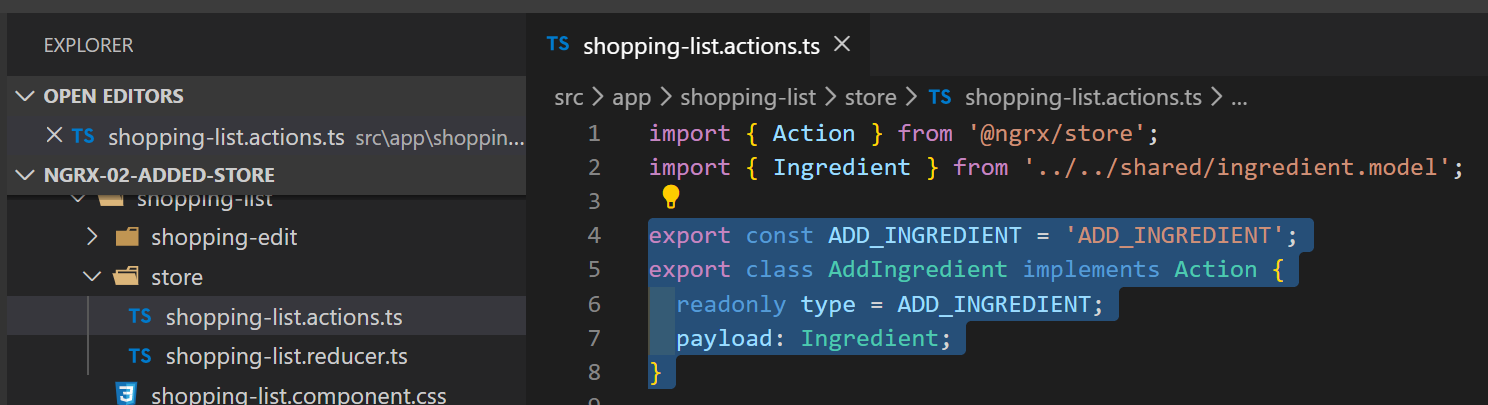
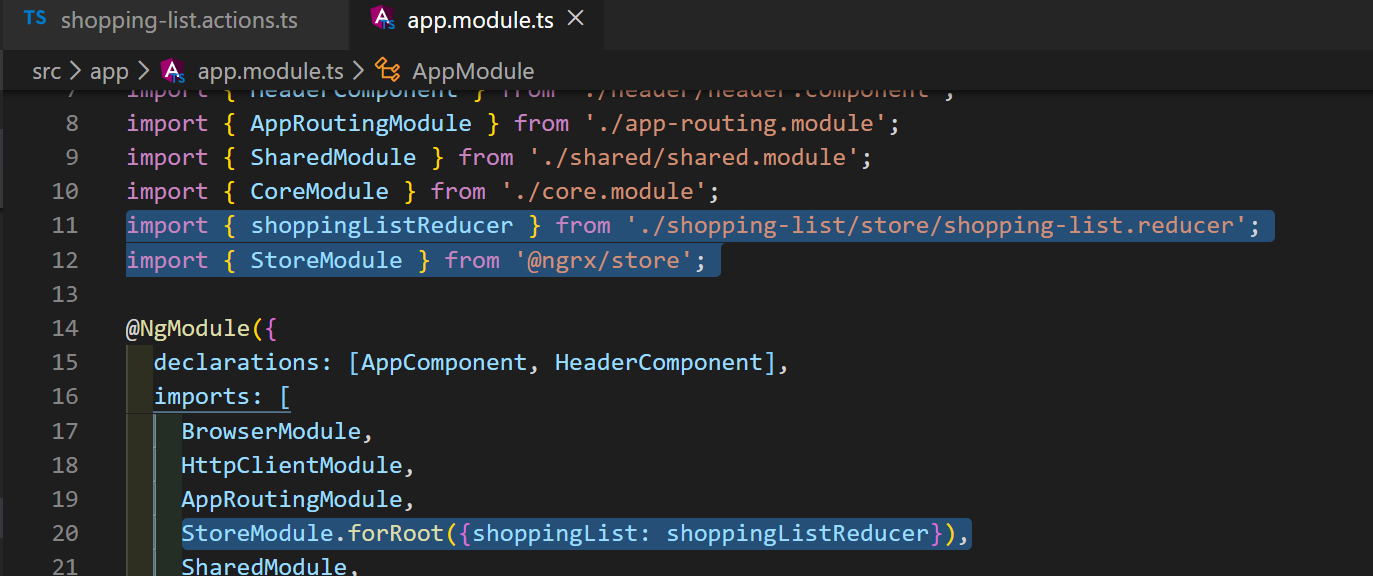
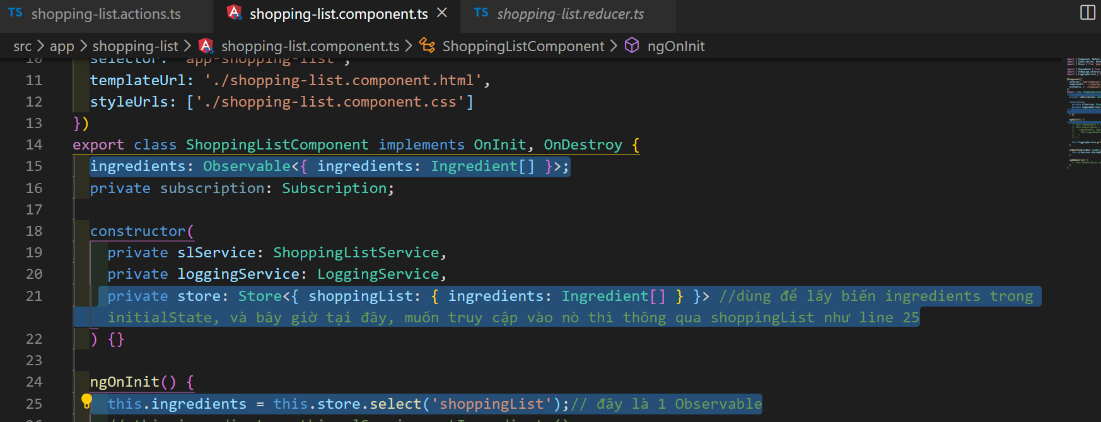
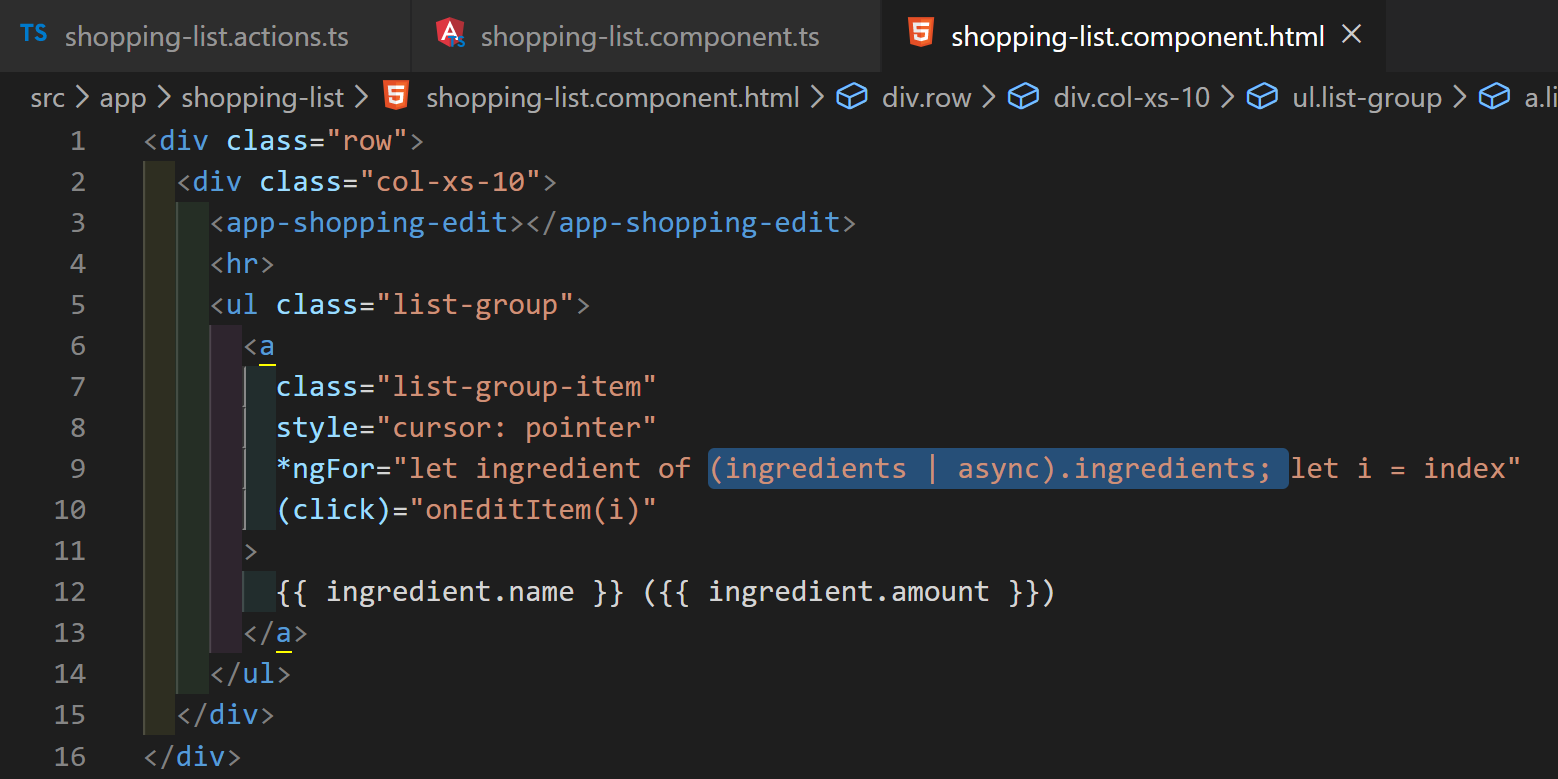
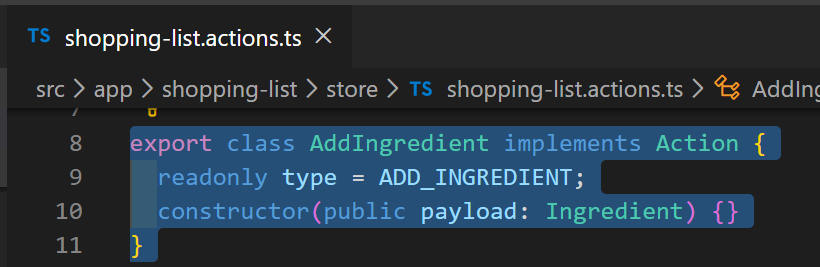
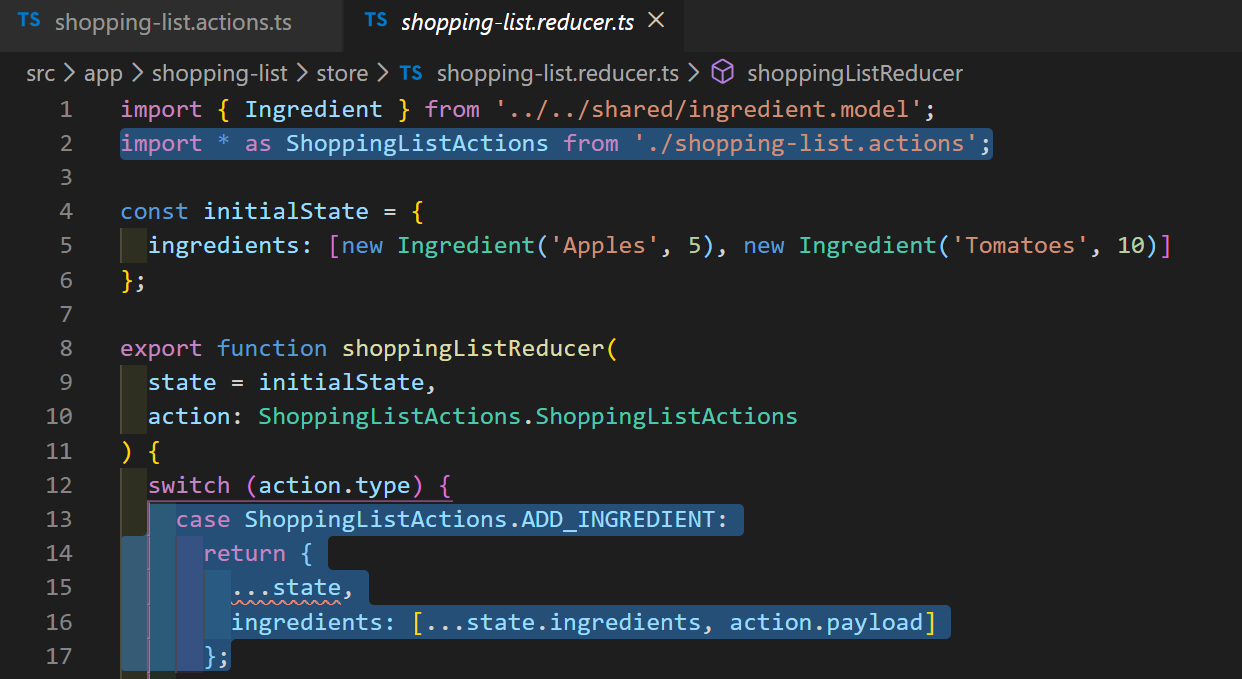
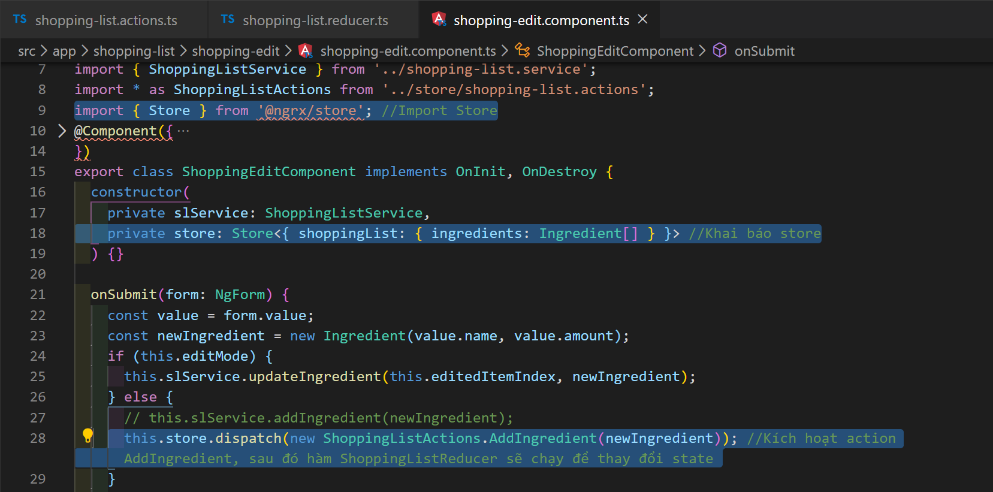
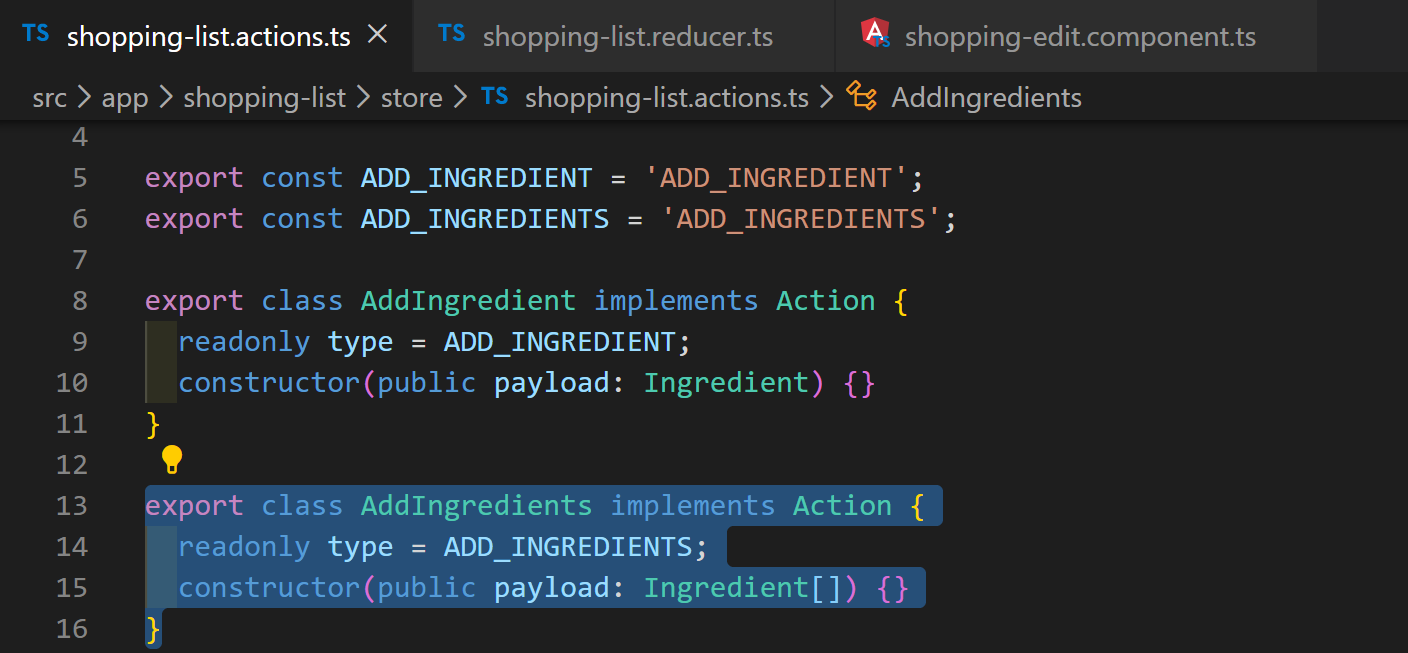
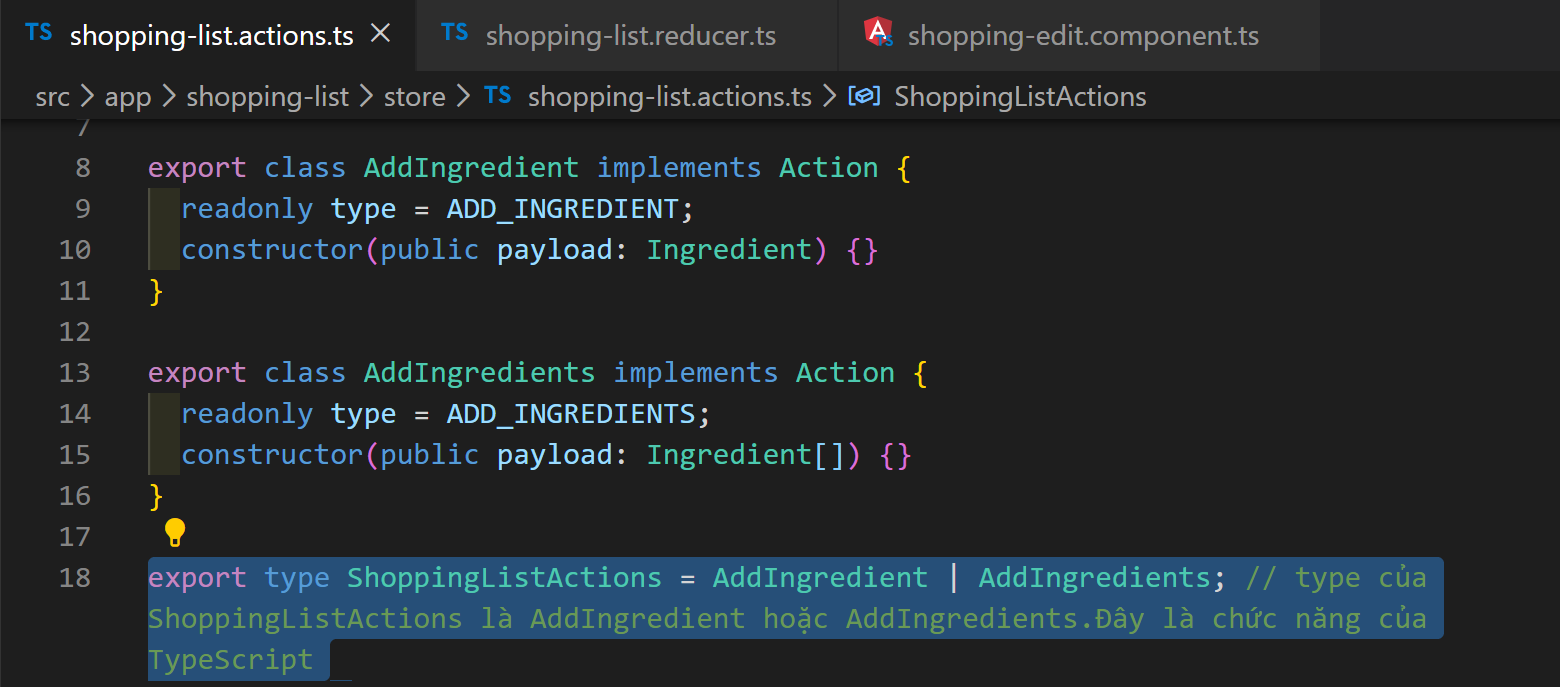
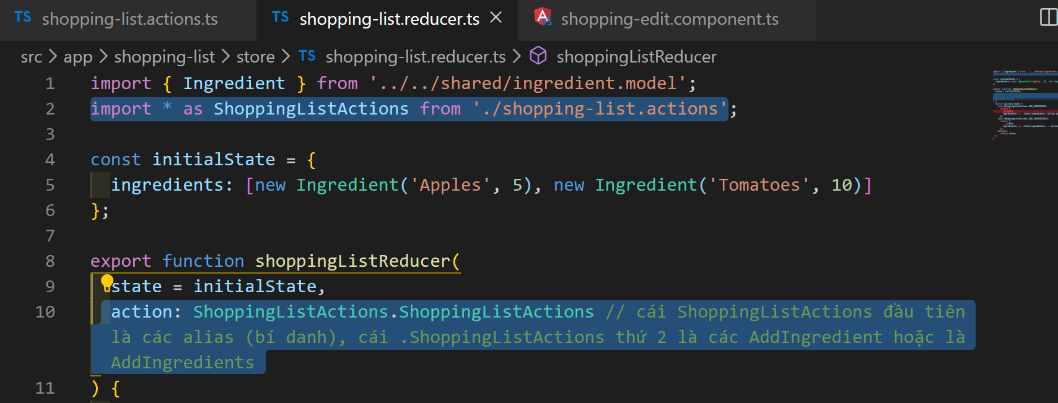
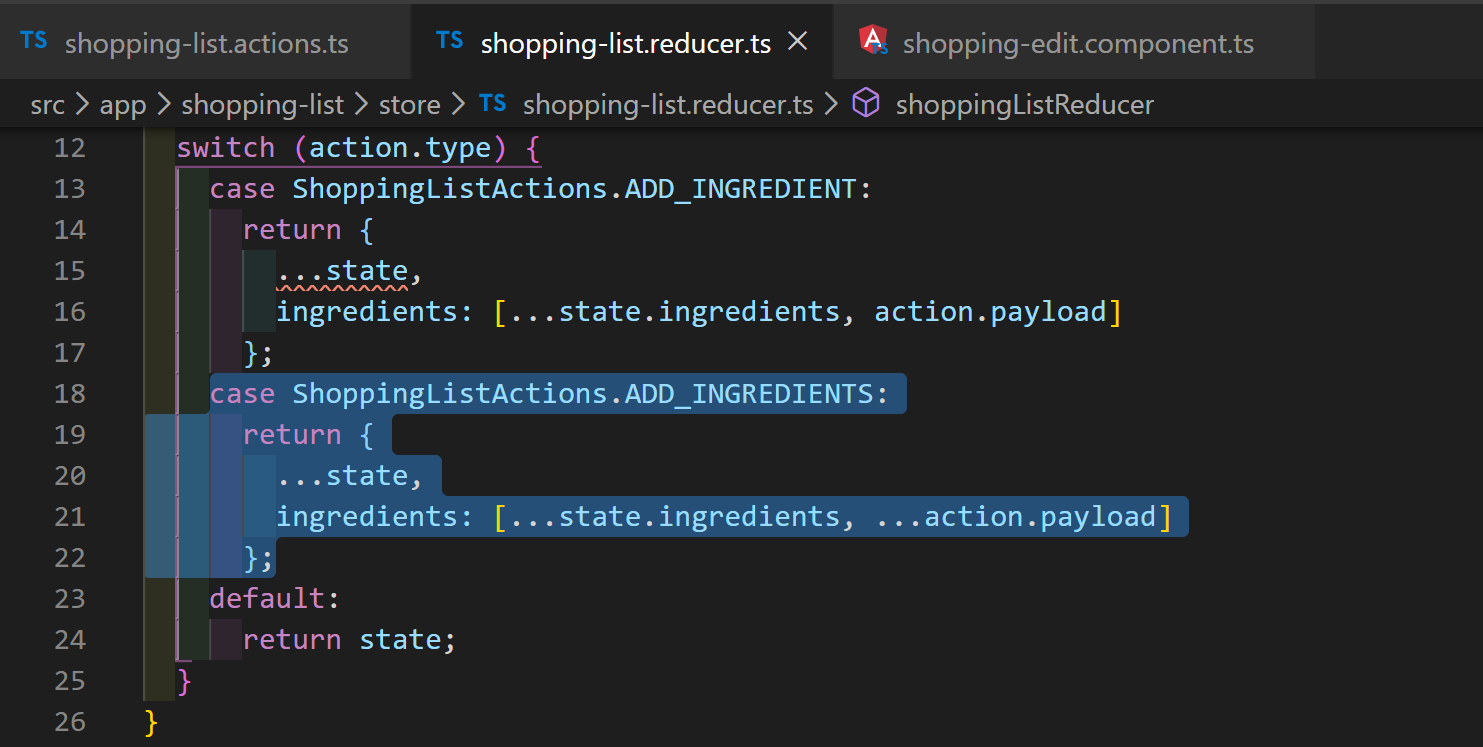
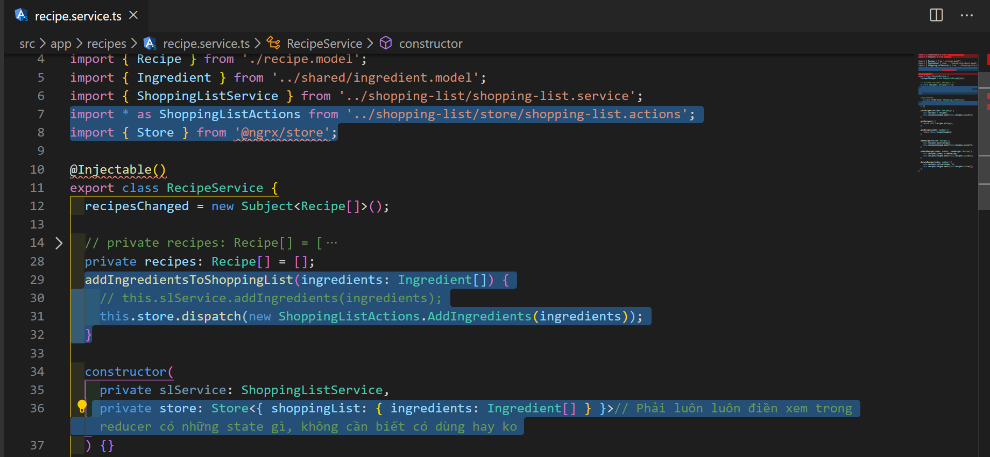
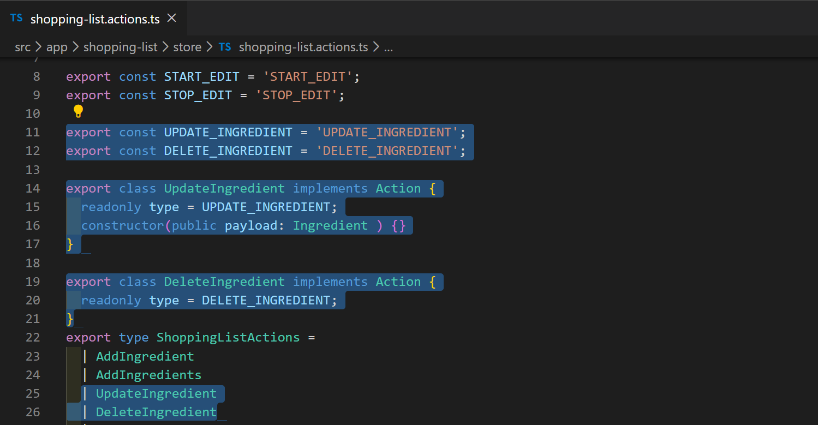
SECTION 24: Working with NgRx in our Project

1. **Getting Started with Reducers:** cài đặt, khởi tạo reducer
   1. 
2. **Adding logic to the Reducer:** Đây là cấu trúc chuẩn của reducer
   1. - Thay đổi state ingredients, thêm Ingredient vào state.ingredients thông qua action.payload.
3. **Understanding and Adding Action:**
   1. Tạo folder Store để chứa các tập tin liên quan đến store như reducer, action.
   2. - Type thì bắt buộc có của Action, còn payload thì tùy vào mình đặt tên, setup.(Như ví dụ trên là đang truyền thêm vào 1 ingredient)
4. **Setting up NgRx store:** 
   1. 
5. **Sellecting State:** Nghĩa là truy cập vào state của store từ component bất kỳ.
   1. - Giong như bên redux
   2. - vì property ingredients là 1 observable, không thể hiển thị như bth được, mà phải làm như trên (line 9).Hoặc ta vẫn subscribe() như bình thương, nhưng ko nên vì dễ dẫn đến leak memory, bug.
6. **Dispatching Actions:** Giong như truy cập vào State của store, bây giờ ta sẽ truy cập, kích hoạt 1 action của store để thay đổi state từ 1 component bất kỳ:
   1. Gỉa sử giờ ta kích hoạt hàm AddIngredient() của ShoppingListReducer:

 -Đây là action AddIngredient.**Nếu có truyền thêm tham số thì viết như thêm. Chỉ có Type là viết ngoài.**

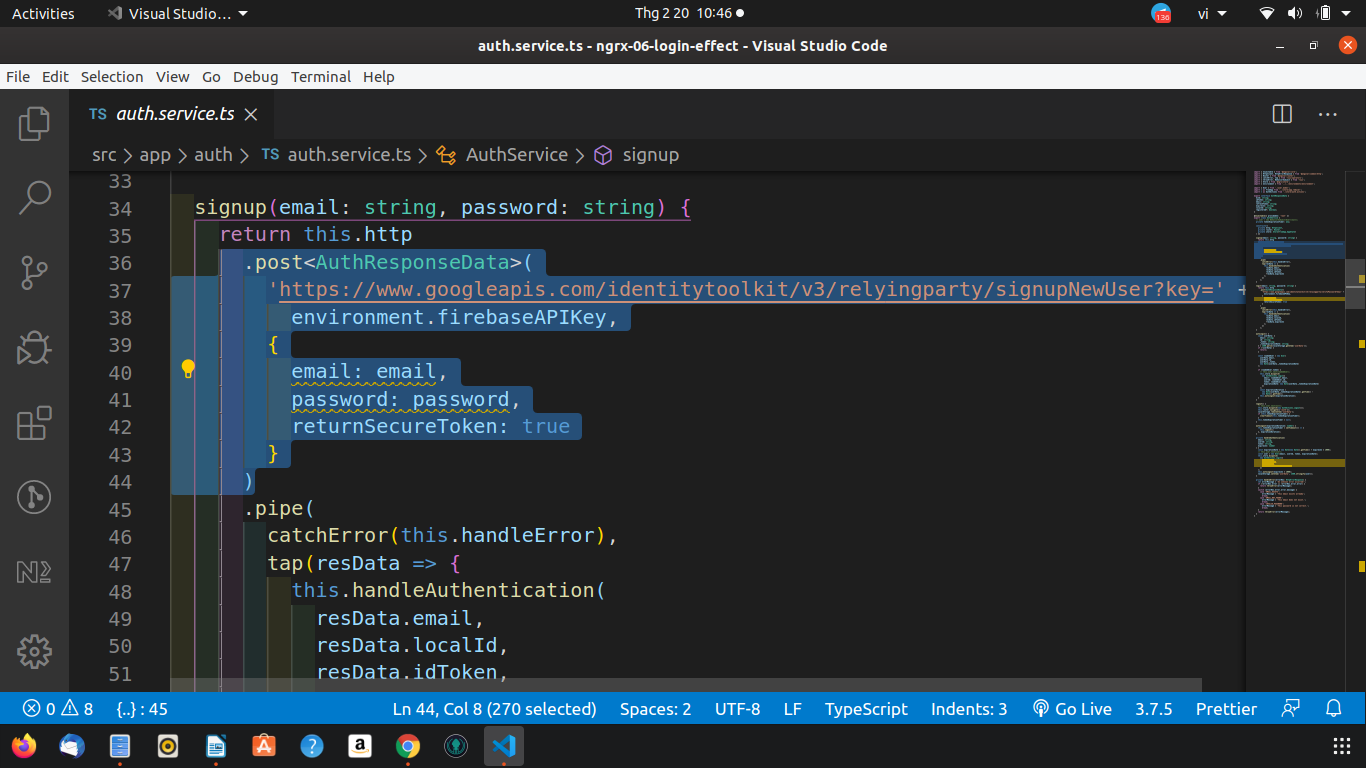
- Đây là shoppingListReducer(), trong đó có action AddIngredient.

-Kích hoạt, chạy action AddIngredient để thay đổi State.

1. **Multiple Actions:** Cách thêm nhiều Actions vào trong 1 reducer:**Thêm Chức năng AddIngredients() vào store, sau đó cấu hình, chạy, kích hoạt, dispatch():**
   1.  -Thêm 1 action AddIngredients vào trong file ShoppingListActions.ts
   2. 
   3. - cái .ShoppingListActions thứ 2 là các AddIngredient hoặc là AddIngredients của 8.2
   4.  -Giai quyết action AddIngredients() logic
   5. -Import, inject store vào Component, sau đó thực thi hàm AddIngredients().
2. **Preparing Update() and Delete() actions:** Các bước chuẩn để viết khai báo 1 action cho Store để thực thi (viết hàm Delete() và Update()), Muốn viết hàm gì cứ làm tương tự các bước sau:
   1. **Khai báo trong actions:**  -Khởi tạo Constant UPDATE,DELETE. Khai báo actions Update, Delete. Export type Update, Delete (line 25 26). **Trong file kia**.

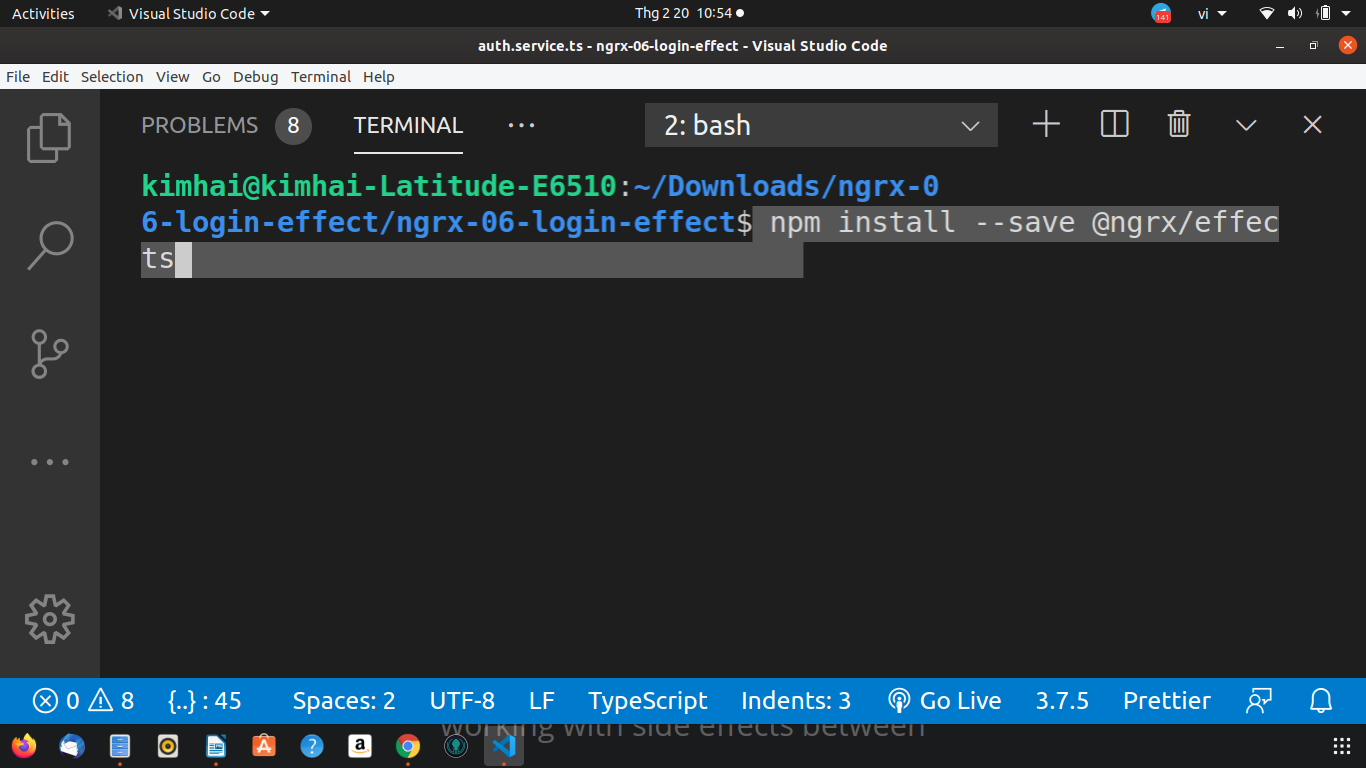
9. **Side Effects:**

**9.1**. **What:** Là những phần code quan trọng cho chương trình nhưng không quá quan trọng trong việc immediate update òf the current state.

9.2. Ví dụ:

Ví dụ như đoạn HTTP trên, dù quan trọng cho chương trình nhưng nó ko ảnh hưởng đánh việc cập nhật lại giao diện. vậy nó là side effect code. hoặc LocalStrorage() cũng là 1 side effect code.

9.3. **Install:**



**10. Defining the First: adding a Login Side Effect**

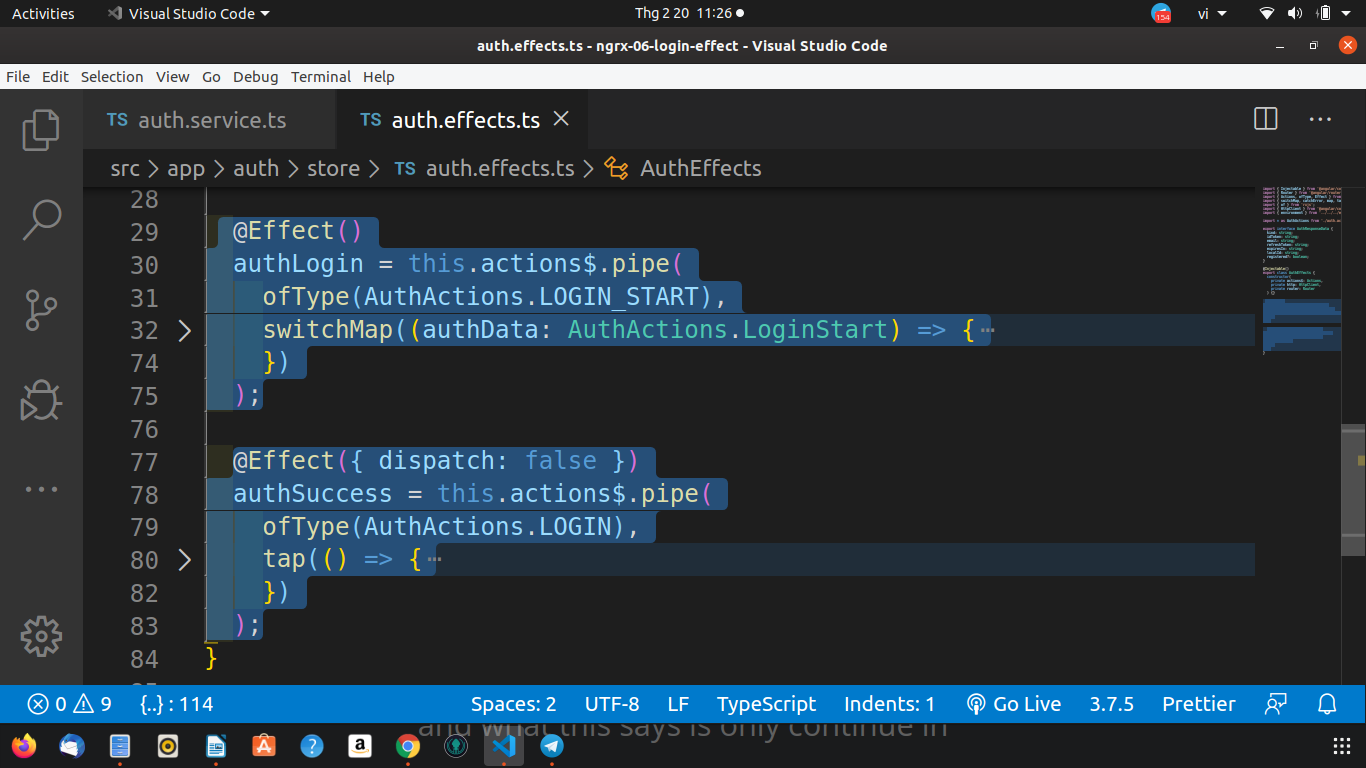
**10.1.** tạo file auth.effects.ts trong folder authStore.

10.2. ./Auth.effects.ts:

-Tạo class AuthEffects.

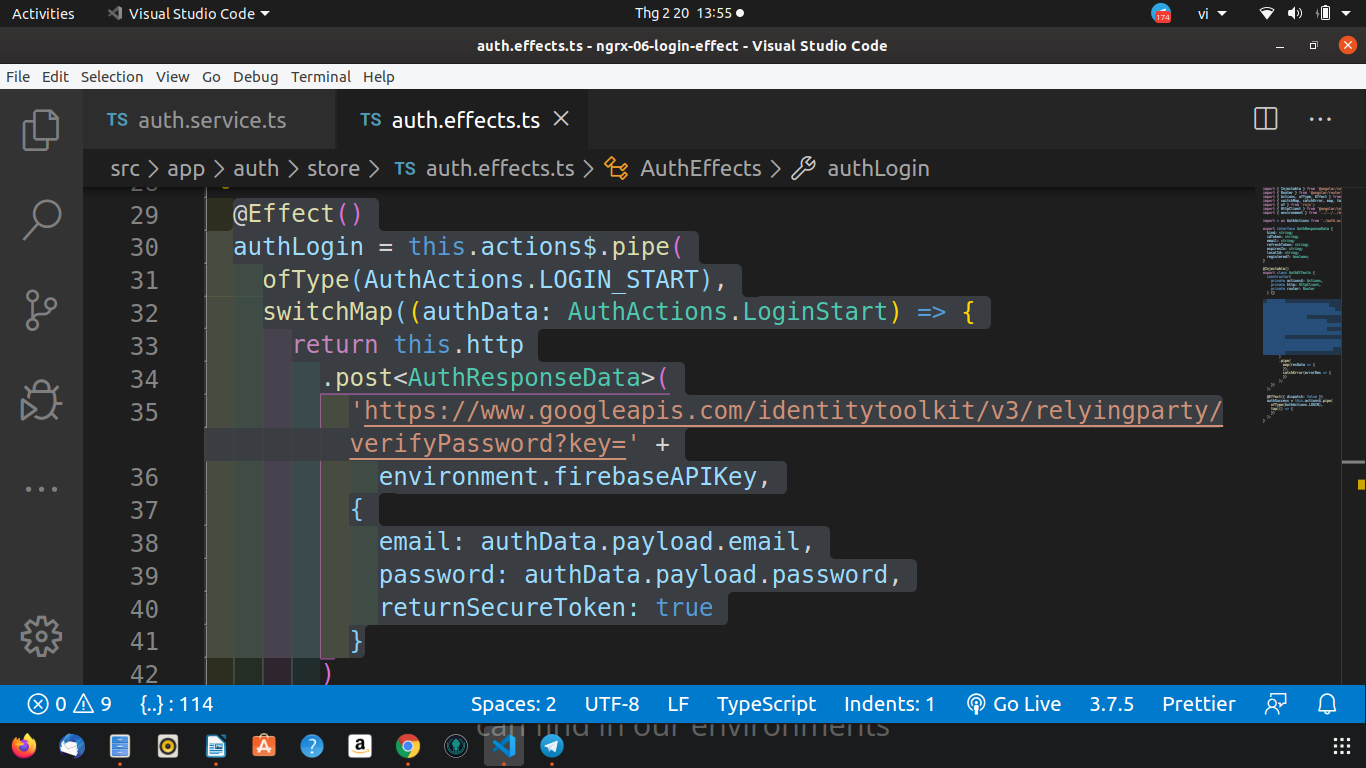
-import Actions$, Ký hiệu $ để báo đây là 1 Observable. (Không bắt buộc).

-**’ofType’ operator** cho phép define a filter for which types of effects you want to continue in thís observable pipe you’re creating. (Nghĩa là nó xác định Dạng của các Effects).



-Nghĩa là effect ‘authLogin’ chỉ chạy khi Action được kích hoạt có cùng type với LOGIN\_START. Các effects khác sẽ không được kích hoạt. Tương tự như với LOGIN.

-LOGIN\_START chính là dùng cho đoạn code <post> HTTP.

-nhớ thêm và import decoration @Effect().

-dùng ‘**SwitchMap’** operator: allow us to create a new Observable by taking another observable’data.(nghĩa là trả về observable HTTP sau khi nhận vào email, username…..).

-First òf all, an effect by default awayls should return a new action at the end when it’s done because thís effect itself doesnot change the state ,it just executes some code.

-Ta phải tiếp tục chức năng nếu như ta không có 1 Error nào từ HTTP request.Do đó ta phải có operator ‘**catchError()**‘. An observable completes whenever an error ís thrown