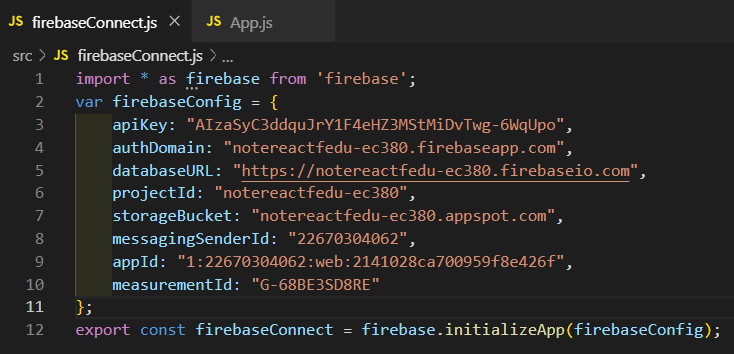
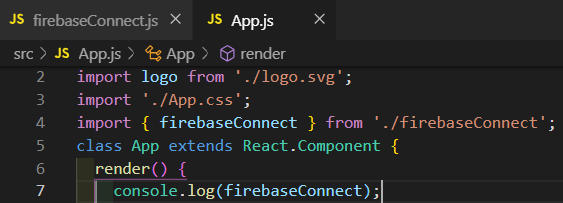
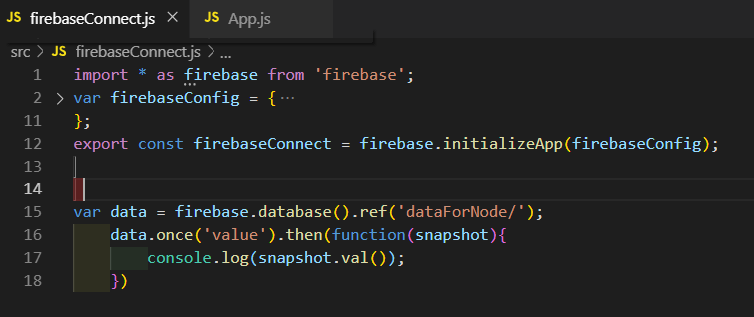
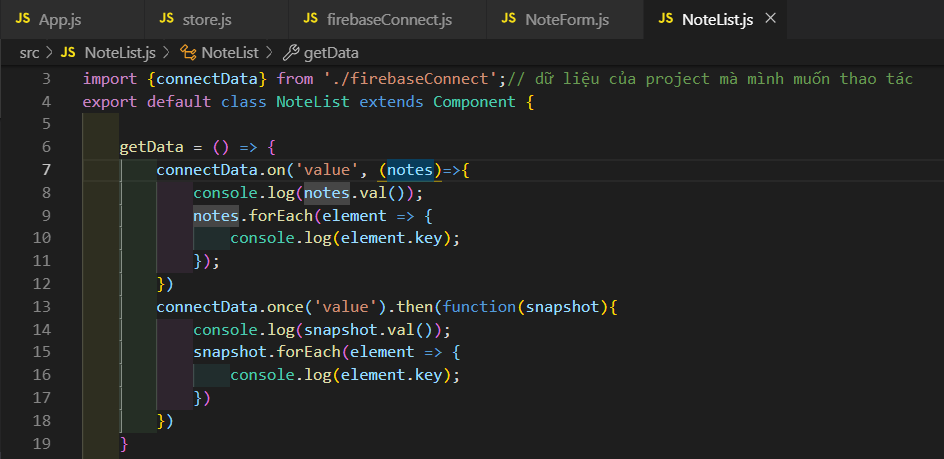
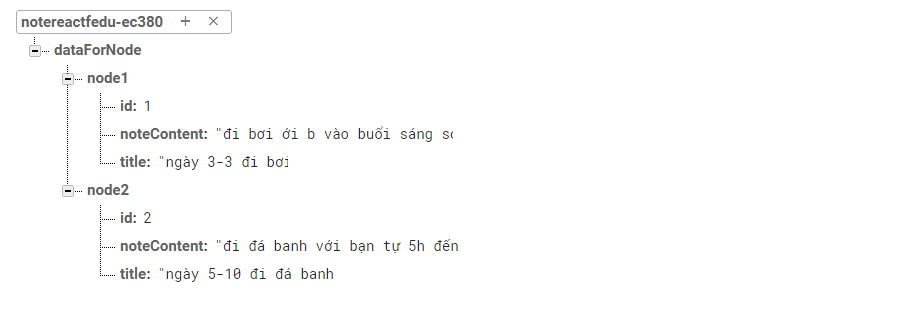
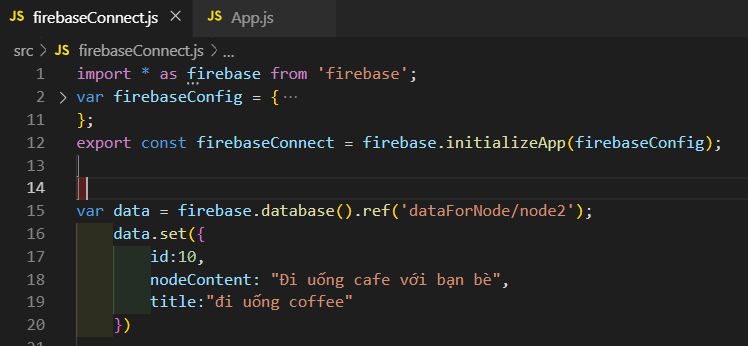
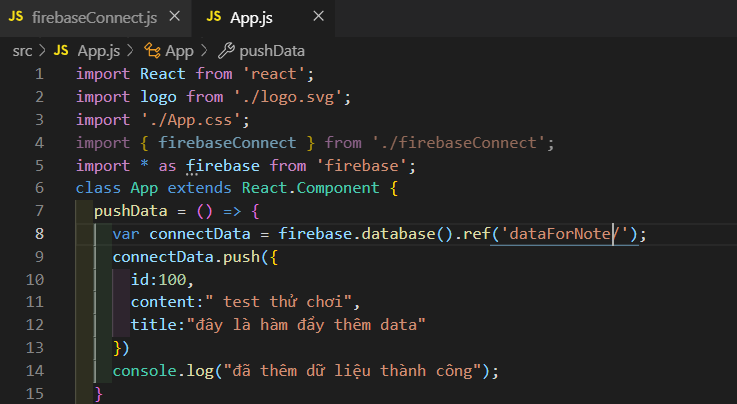
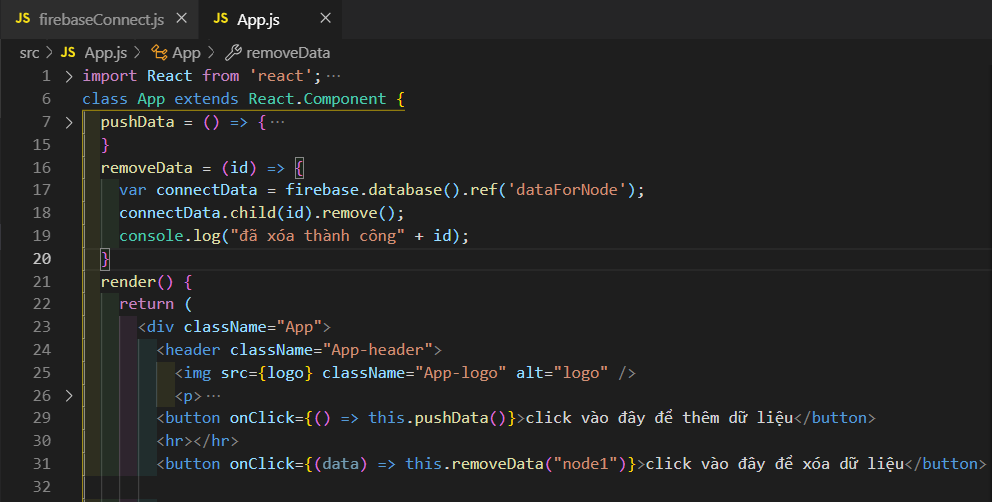
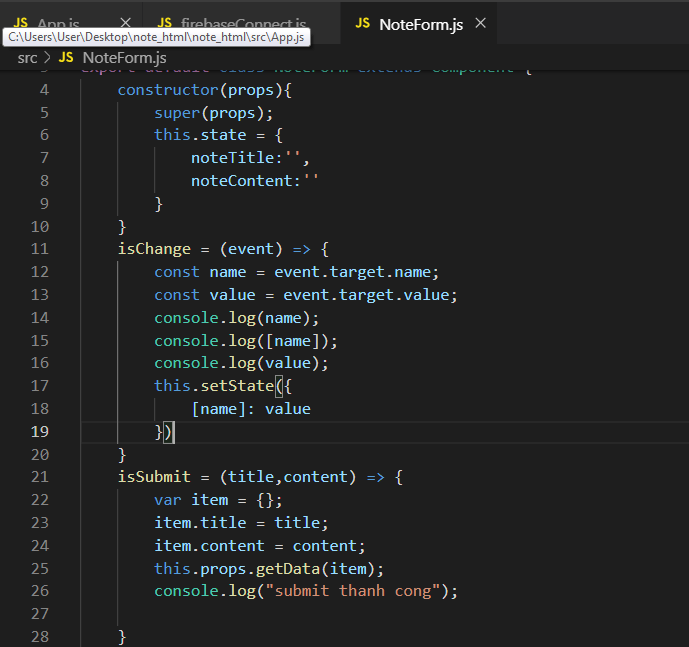
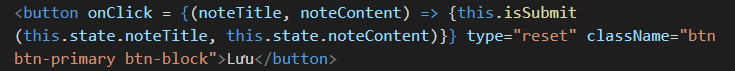
sCÁC CHỨC NĂNG:

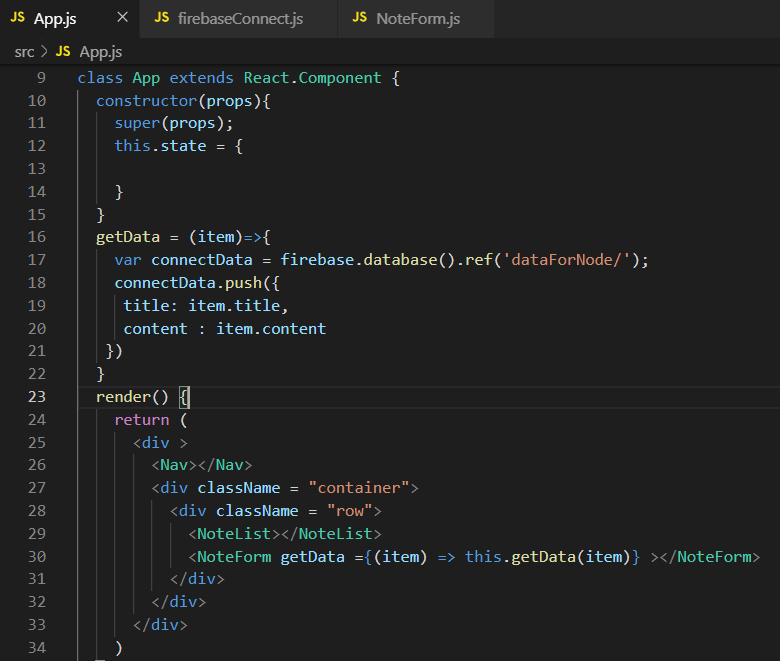
* Thêm, xóa, sửa,xem.

**Tạo giao diện úng dụng .**

* 1.Tạo giao diện html.
* 2.Convert sang Component.
* 3.Tạo Store và kết nối.
* 4.Viết logic xử lý từng chức năng.

**Cơ sở dữ liệu FIREBASE:**

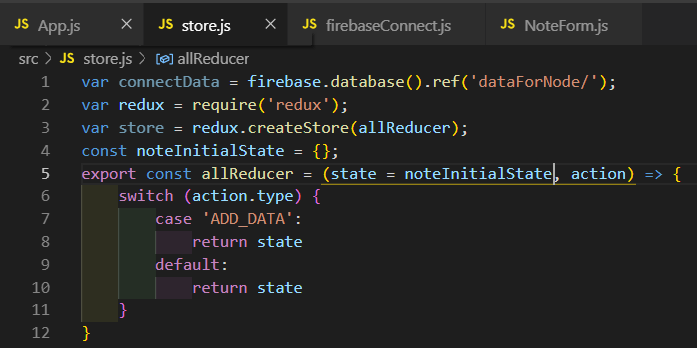
* WHY:
  + Tối đa 5G.
  + Realtime.
  + Cơ sở dữ liệu trực tuyến => Không cần cài đặt nhu SQL, MySQL.
  + Cũng không cần API
  + ReactJs giao tiếp trực tiếp với firebase
  + Chỉ dùng cho các web, app nhỏ, các web lớn có nhiều chức năng thì không làm được
* HOW: <https://firebase.google.com/>
  + Tạo project:
    - <https://console.firebase.google.com/u/0/?hl=en>
    - Realtime
    - Export JSON
  + Npm install firebase
  + Các lệnh cơ bản của firebase:
    - Thêm mới dữ liệu thông qua reactJs(trực tiếp)
    - Lấy dữ liệu
    - Sửa dữ liệu
    - Xóa dữ liệu
* ***KẾT NỐI REACTJS – FIREBASE***:
  + Tạo CSDL trên firebase.google.com và copy mã script.
  + Tạo app reactjs.
  + Cài đặt modult ‘npm install firebase –save’.
  + Tạo file ‘firebaseConnect.js’.
  + Trong file ‘firebaseConnect.js’, copy và past đoạn script trên firebase.google.com:
    - 
  + Ở ‘App.js’:
    - Import Firebase:
    - 
* ***CÁC CHỨC NĂNG CƠ BẢN:*** lấy, sửa, thêm, xóa….
  + **LẤY DỰ LIỆU** CỦA BẢNG DỮ LIỆU BẰNG ‘DataSnapShot’:
    - Dữ liệu của firebase được lưu trong phần tử gọi là dataSnapShot(ảnh chụp dữ liệu)
    - Do đó, đầu tiên phải khai báo đối tượng dataSnapShot trước:
      * 
      * HOẶC DÙNG HÀM on(): đây là kiểu callBack
      * 
      * Cả 2 cách trên điều được nhương On là callback, Once() là promise.
    - 
  + **SỬA DỮ LIỆU** TRONG FIREBASE: dùng hàm set
    - 
  + **THÊM DỮ LIỆU** TRONG FIREBASE:
    - Gồm 2 cách: hàm set() như trên, hoặc dùng hàm push().
    - Hàm Set() thì phải tự đặt id, nó ko hỗ trợ tự tạo id.
    - Hàm Push() hỗ trợ tự đặt id (id do firebase cung cấp) nên tiện dụng hơn:
      * 
  + **XÓA DỮ LIỆU** TRONG FIREBASE:
    - Dùng hàm .child(data).remove():
    - 
* **THỰC HÀNH ỨNG DỤNG GHI CHÚ:**
  + Chức năng **THÊM MỚI** vào cơ sở dữ liệu:
    - Gồm 3 bước:
      * 1.Kết nối CSDL(làm như trên).
      * 2.Click nút “lưu” thì lấy thông tin của dữ liệu vào:
        + 
        + 
        + 
      * 3.Lấy được dữ liệu thì update vào CSDL, gồm 2 cach:
        + Không sử dụng redux (reactjs thuần)



* + - * + Sử dụng redux:



1. Tạo Store (file ‘store.js’):Cài đặt redux và khởi tạo reducer

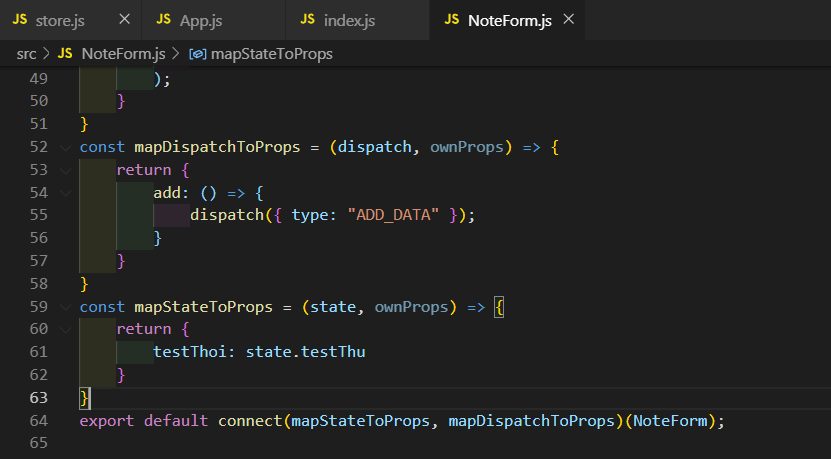


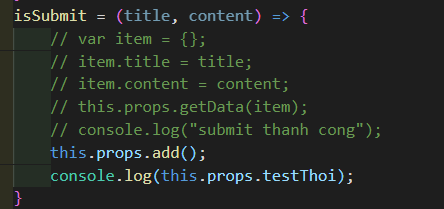
2.Kết nối Store với component <NoteForm>:

Gồm 3 bước cơ bản:

Thực hiện kết nối Store với component như mình đã học.

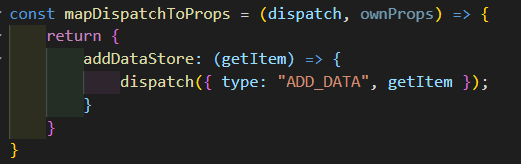


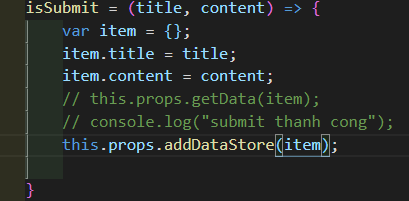


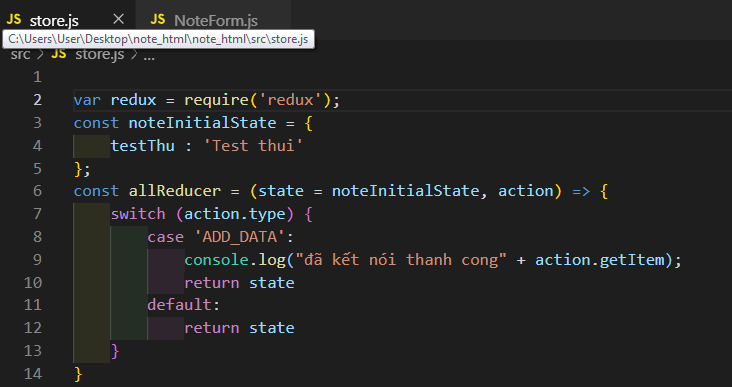


Truyền tham số vào Store và xử lý tham số ấy trong store:

Truyền tham số vào các Reducer(các chỉ thị, thực thi) ở trong store với cú pháp thực thi là: action.tenThamSo .

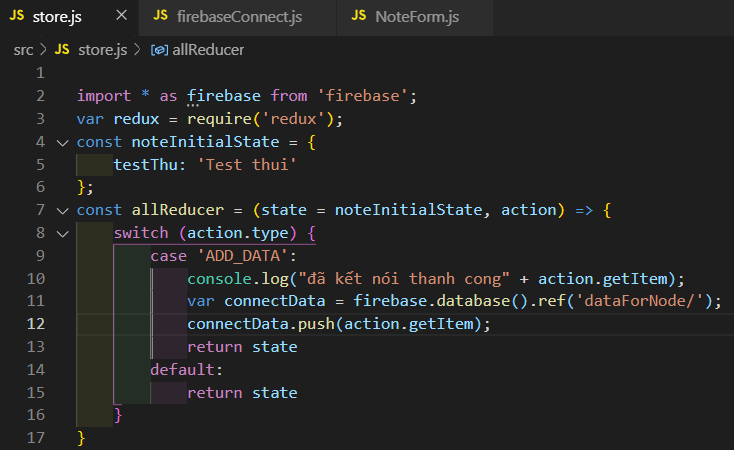


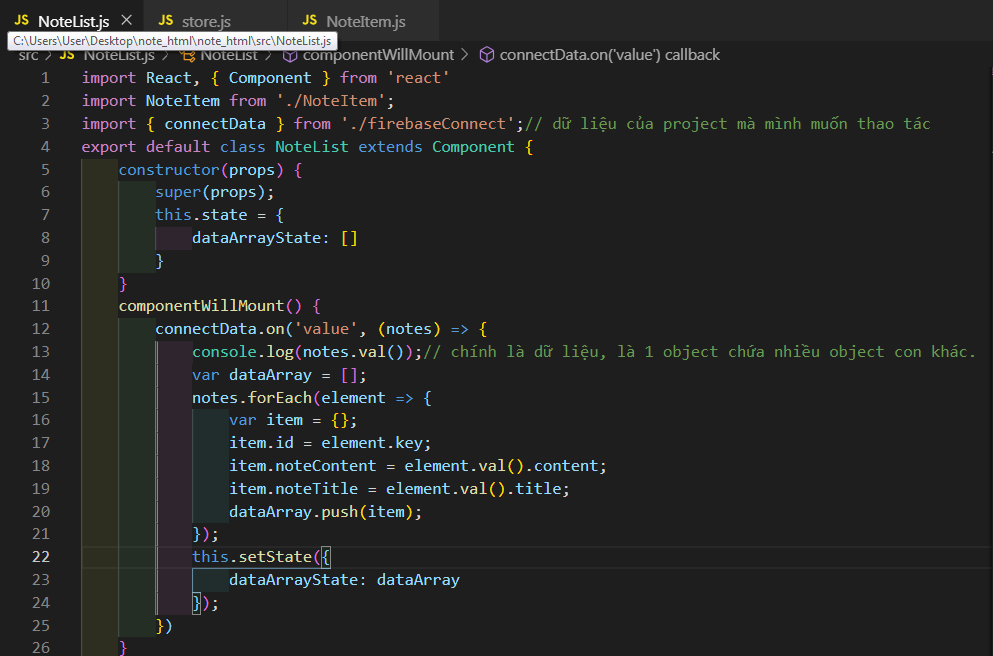


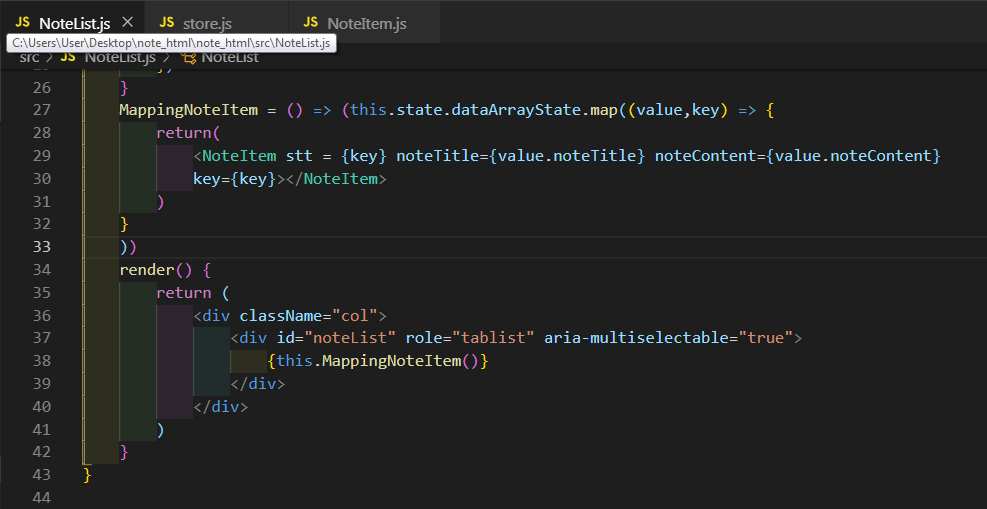


Hoàn chỉnh chức năng THÊM MỚI data:

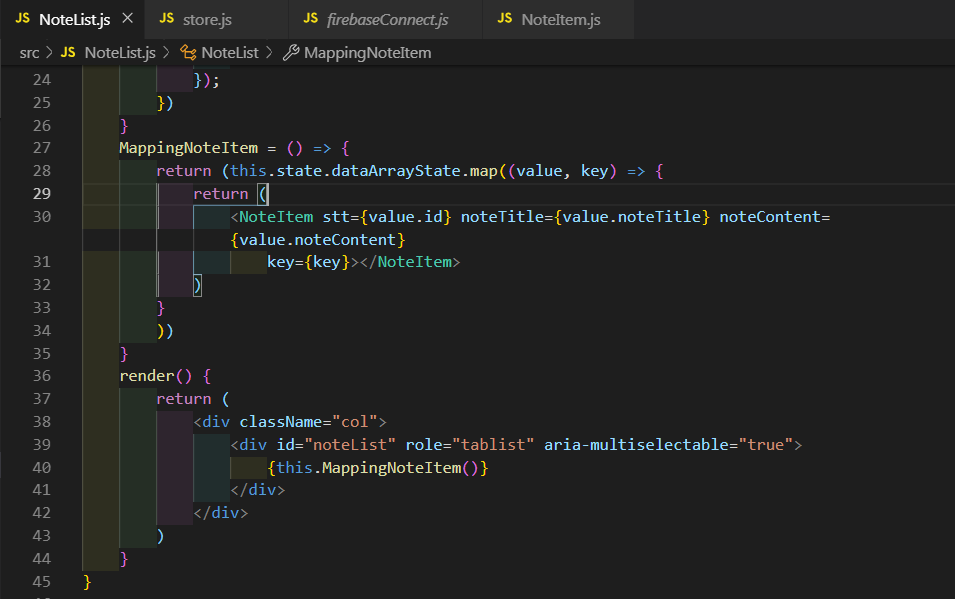
Sau khi store nhận được dữ liệu thì dùng hàm Push() đẩy item lên CSDL Firebase



* + Chức năng LẤY DỮ LIỆU từ CSDL và IN DỮ LIỆU ra ở <NoteList>:
    - Gồm 2 bước :
      * 1.Lấy dữ liệu từ FIREBASE:
        + 
        + ‘notes’ chính là dữ liệu data trên firebase.
        + notes.val() là 1 object chứa nhiều object con khác, vì thế, cần phải biến đổi (convert) object lớn ấy thành các object item con bằng cách trên.
        + Tạo 1 cái mảng ‘dataArray’ và đẩy các đối tượng trên vào mảng bằng hàm .push().
        + Cập nhật lại State bằng mảng vừa tạo.
      * 2.In dữ liệu ra bằng hàm Foreach() hoặc map():
        + Map():



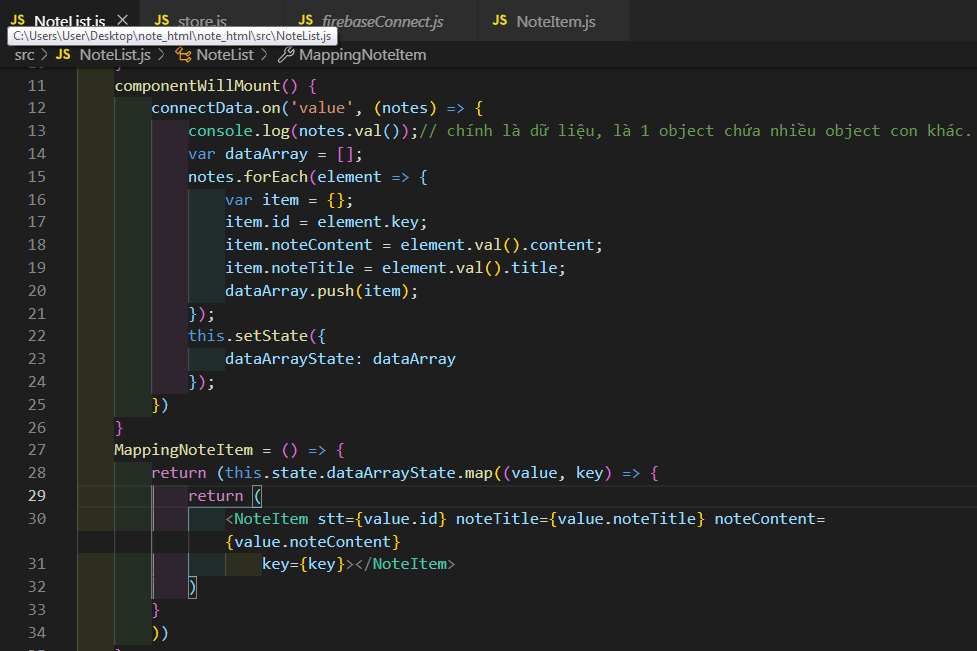
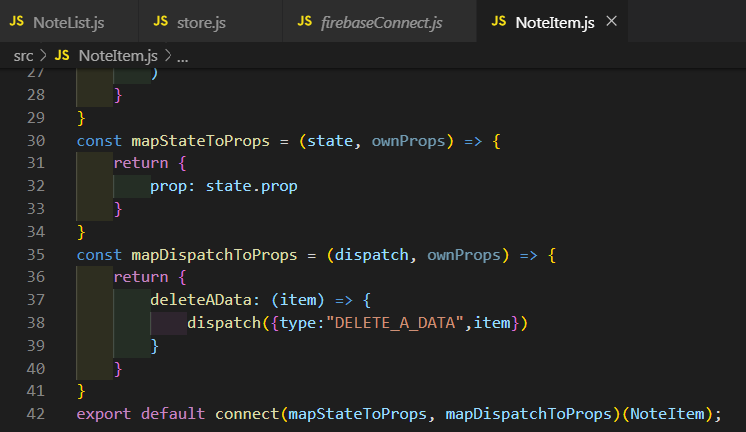
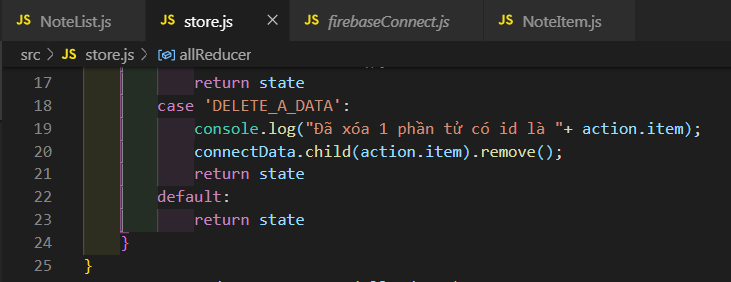
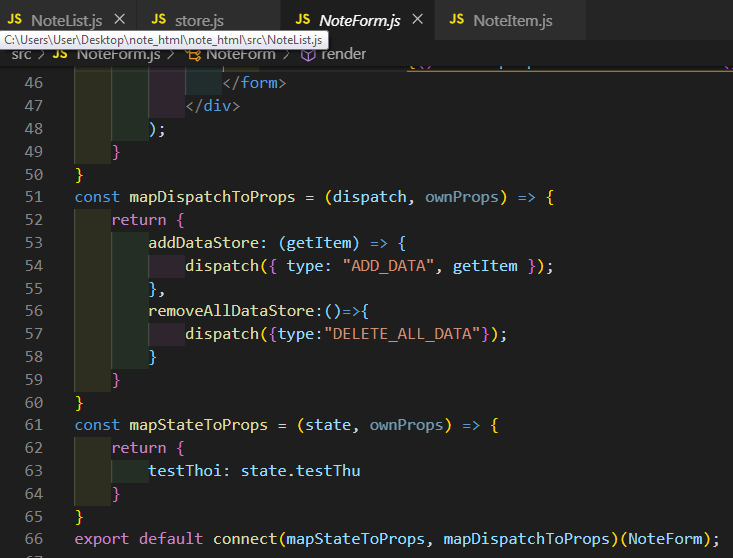
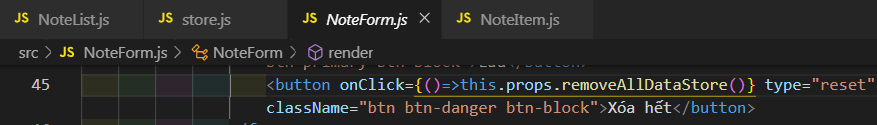
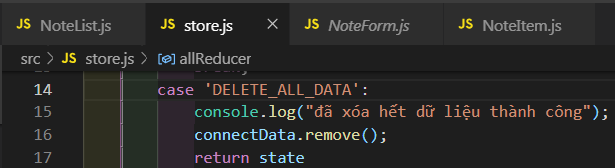
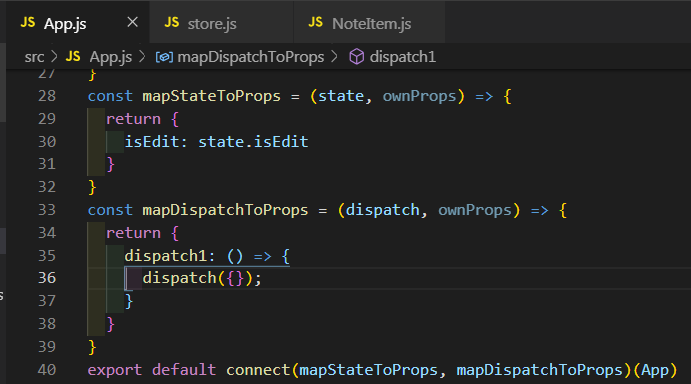
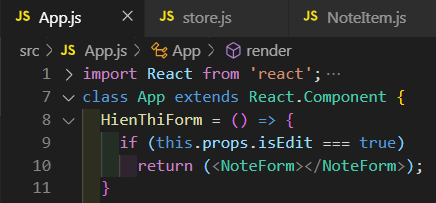
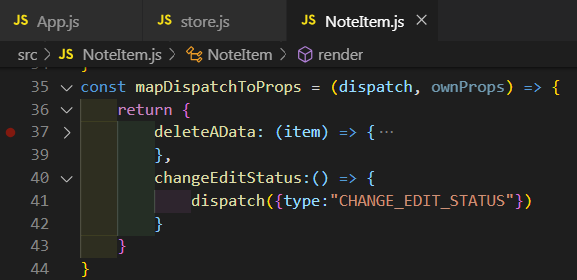
Hoặc:



* + - * + ForEach(): Thực hiện không được vì:

ForEach() không thể trả về (return) bất kỳ giá trị nào. Bất kỳ giá trị nào trả về đều bị loại bỏ (undefined).

<https://viblo.asia/p/so-sanh-foreach-va-map-trong-javascript-L4x5x1YBKBM>

* + Chức năng **XÓA DỮ LIỆU** ở CSDL: xóa 1 dữ liệu và xóa hết dữ liệu:
    - **XÓA 1 NOTE:**
      * ****
      * **Tại <NoteList>, ta lấy** được ‘id’ của từng Note bằng ‘element.key’.
      * **Truyền** ‘id’ của các NOTE xuống <ListItem> bằng props ‘stt’.(vậy là từng NoteItem đều nhận được id riêng của mình như -LtspbTZixU3AjHuNkPi; -Ltspba\_tdC5JvJLuhsT).
      * **Tại <NoteItem>, kết nối với Store bằng các bước**  đã học:
        + ****
      * **Tại** Store.js: Khai báo case “DELETE\_A\_DATA” và nhận vào 1 action.item (id của note đó)
      * Dùng hàm child(id).remove() để xóa Note cần xóa tại CSDL Firebase.
      * ****
    - **XÓA TẤT CẢ NOTE:**
      * ****
      * **Tại** <NoteForm>, kết nối component này với store như trên.
      * Khai báo hàm removeAllDataStore() để xóa tất cả dữ liệu
      * Tại <button> ‘xóa hết’ , onClick để xử lý sự kiện xóa hết dữ liệu:
        + ****
      * ****
      * **Tại** store.js, viết case “delete\_all\_data” để xóa hết tất cả dữ liệu trên csdl FireBase.
  + Chức năng HIỂN THỊ FORM SỬA khi click vào nút “Sửa”:
    - ****
    - Kết nối store với <App> và <NoteItem> bằng các cách đã học (như trên).
    - Sau khi kết nối với Store:
      * ****
      * Tại <App>, ta nhận được this.props.isEdit. Từ đó viết arrow function HienThiForm()
      * ****
      * ****
      * **Tại <NoteItem>, ta nhận** được this.props.changeEditStatus() để thay đổi State tại Store.