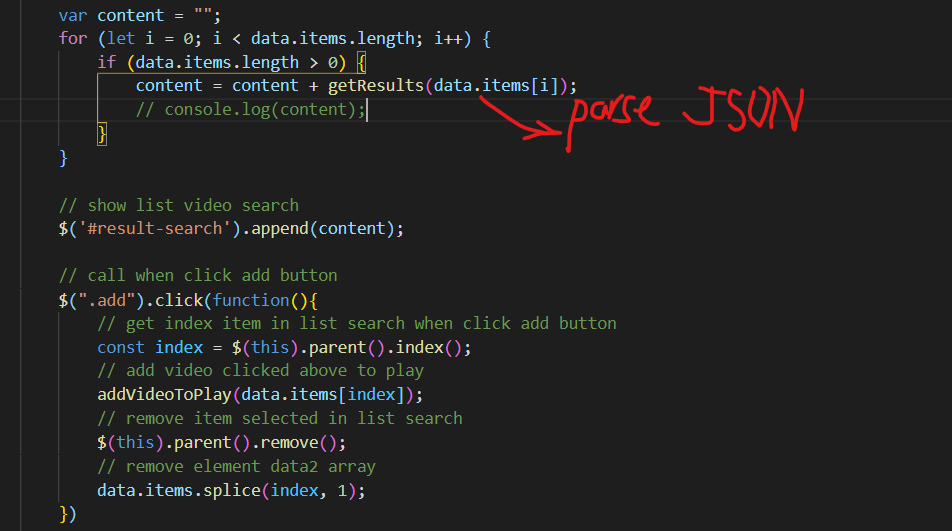
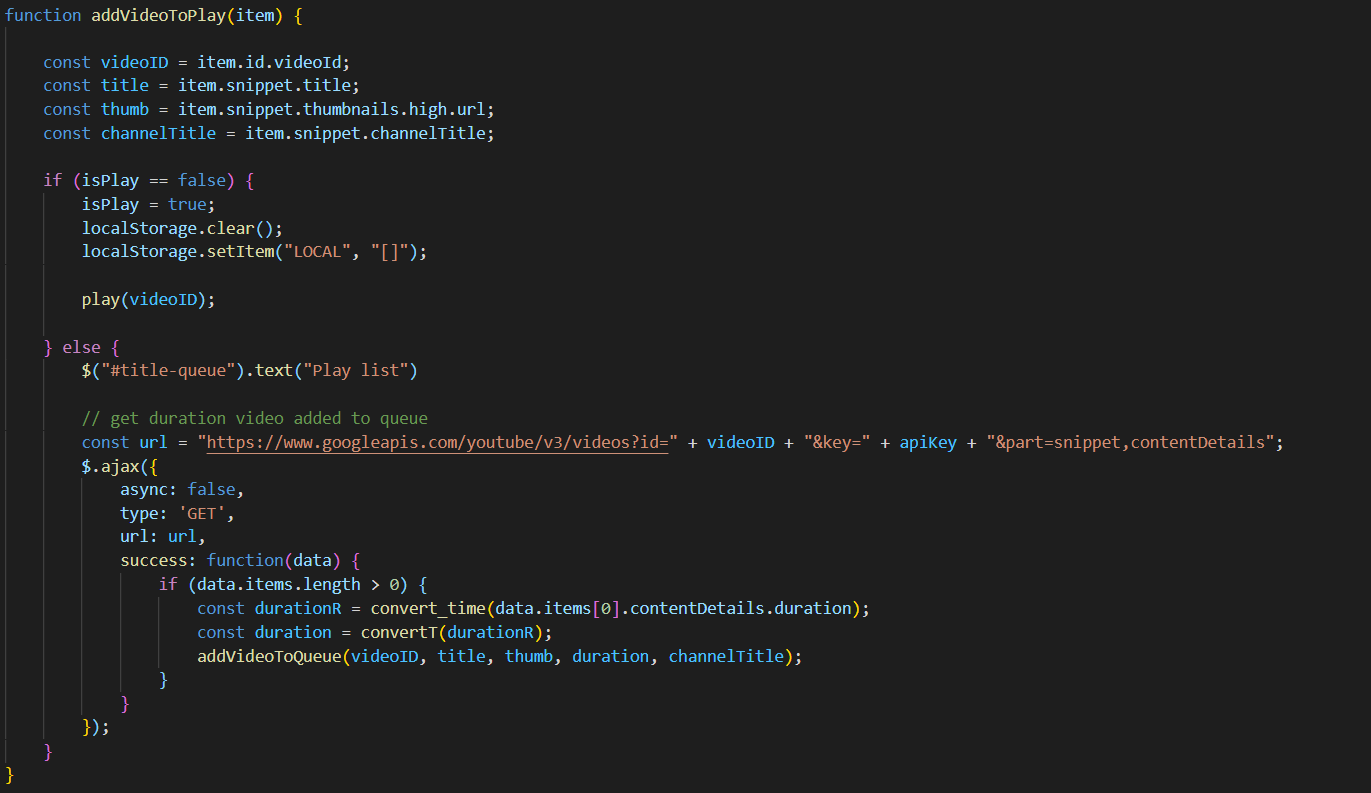
Link github: [thanhtinhutc2/youtube-api (github.com)](https://github.com/thanhtinhutc2/youtube-api)

Các công nghệ sử dụng:

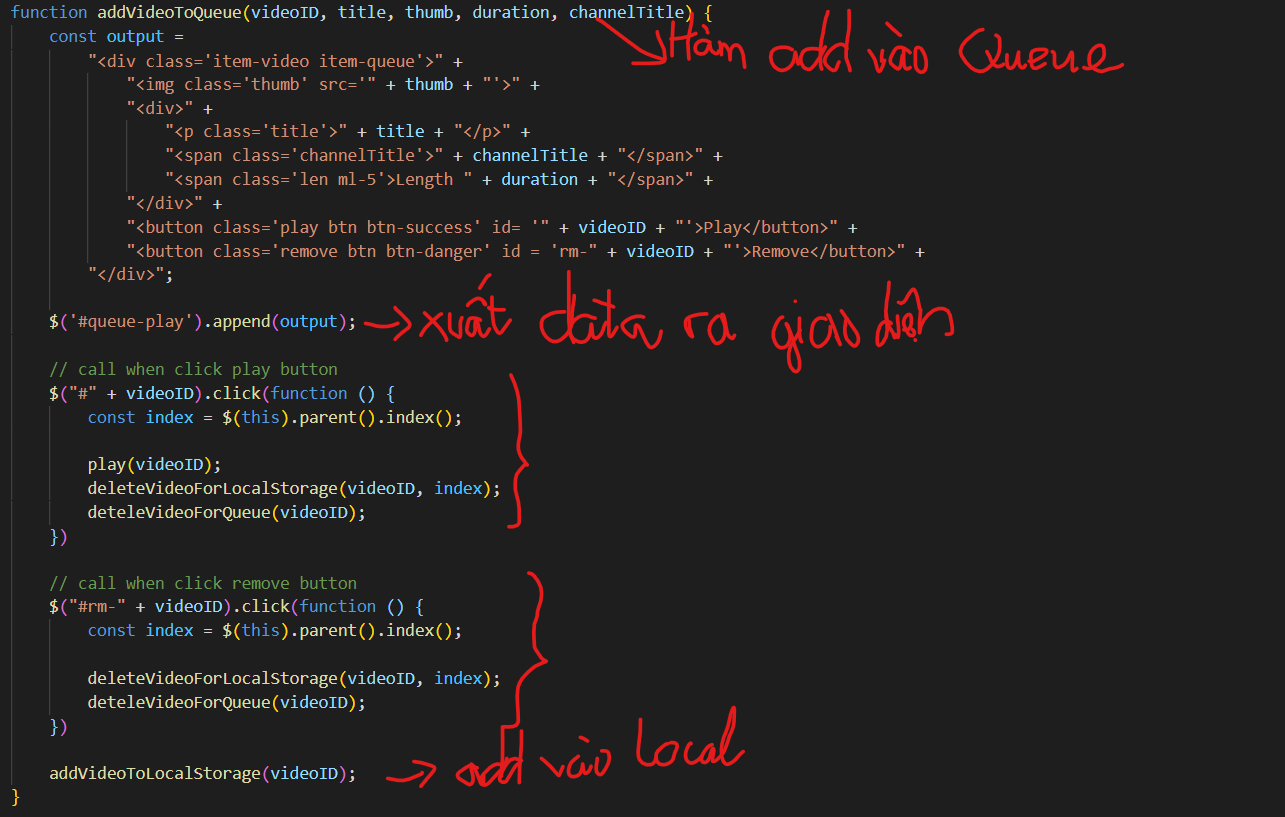
* Giao diện:
* Sử dụng Bootstrap4, Materialize để tạo giao diện.
* Kết hợp với Javascript để lấy dữ liệu năm hiện tại cho Footer và nút Back to top khi người dùng scrool xuống dưới.
* Lấy data từ Youtube API:
* Sử dụng AJAX để get data từ Youtube.
* Sau khi có data từ API, ta sẽ lấy các data sau để render giao diện: videoID (để gán vào link nhúng iframe của youtube), title, thumbnails, channelTitle.
* Sau khi có data đc gọi từ youtube api thì parse vào file JSON để hiển thị dữ liệu vừa call đc ra giao diện.



* Hàm splice( , ) dùng để xóa một phần tử ra khỏi danh sách PLAY LIST khi nhấn vào nút remove có id=”rm-”+videoID. Có nghĩa là vị trí index sẽ được thay bằng id = “rm-” + videoID.
* Khi click nút Add to playlist thì nó sẽ gọi ra vị trí của item đó ra đồng thời show ra giao diện bên phải có id=”result-search”.

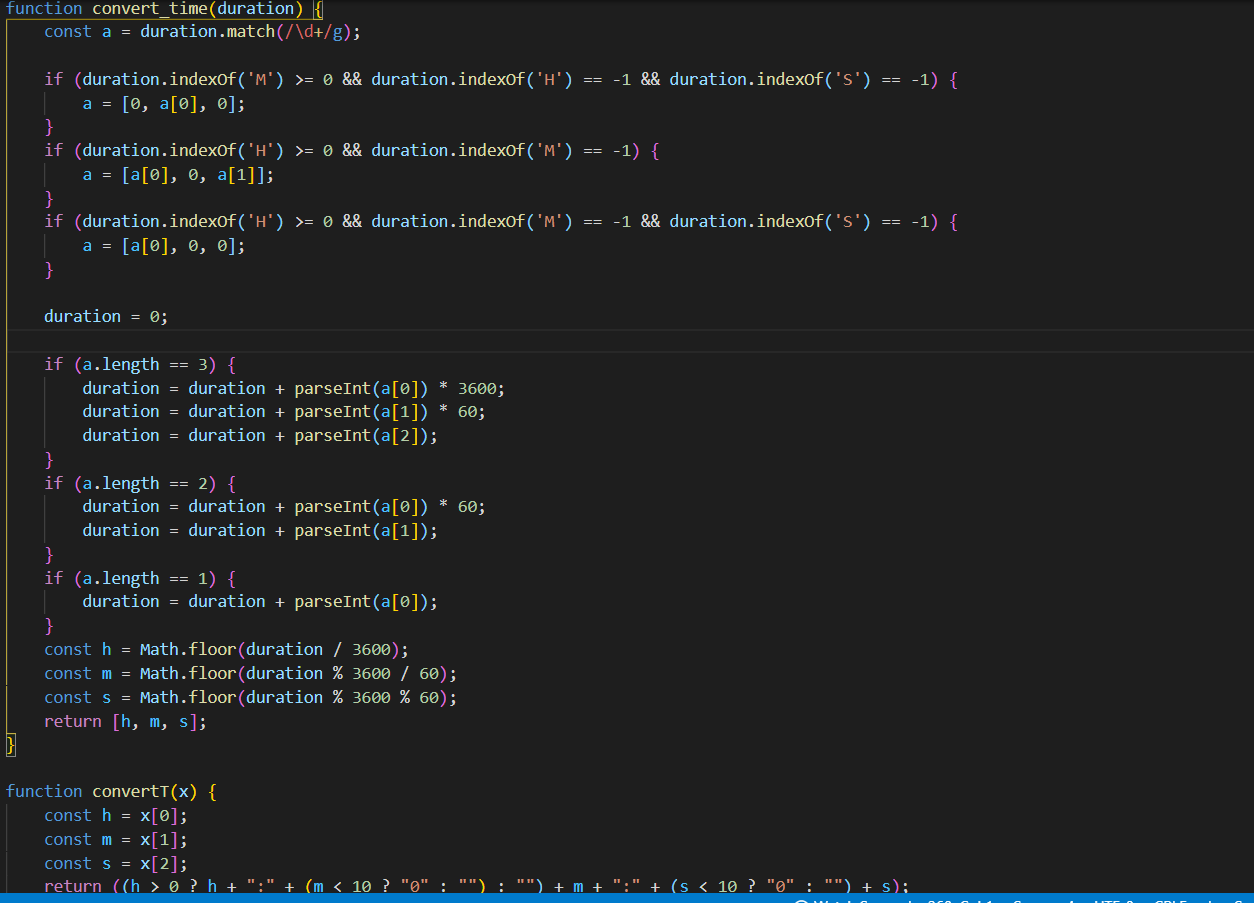


* Khi hàm addVideoToPlay đc gọi thì truyền videoID vào đối tượng player ở trên. Để xử lý các request từ user thì sử dụng cơ chế hang đợi FIFO để add một đối tượng vào hang đợi. Sau đó lưu vào localStorage để lưu lại dữ liệu hang đợi đó để tiết kiệm thời gian thay vì kết nối với SQL.
* Tạo ra một mảng để sử dụng Queue để lưu trữ dữ liệu các item đc add vào Queue.



Khi click vào nút remove nó sẽ xóa data của item đó khỏi localStorage và Queue

Khi click vào nút play nó sẽ lấy id của video đó gán vào link nhúng để phát video, đồng thời xóa khỏi hàng đợi và localStorage

* Hàm convert time: API lkhông trả ra trực tiếp độ dài của video theo dạng milliseconds hoặc string biểu diễn độ dài như "04:56" mà đây là chuẩn Date and time format - ISO 8601. Hàm này sẽ làm chức năng định dạng lại thời gian của video theo định dạng mong muốn.
* 
* 