

Bài thực hành – 06 – AJAX

Mục tiêu:

- Thực hành thao tác AJAX kết nối với API.
- Khởi tạo project ExpressJS để sử dụng AJAX để kết nối đến API.
- Lưu ý tái sử dụng lại các bài tập đã làm trong Lab 05 để không phải xây dựng lại API đã làm ở bài Lab 05 trước đó.

Bài 1: (3 điểm)

- Tạo một project trên Firebase có tên là Product Management.
- Tạo một Firebase Realtime Database có cấu trúc object lưu trữ như sau:

```
{
  "products": {
    "objectID": {
      "name": "Galaxy S 21",
      "price": "1000",
      "brand": "Samsung",
      "category": "mobile"
    }
  }
}
```

- Tái sử dụng lại là bài 1 – Lab 05 để xây dựng API để thực hiện (đã được làm trong La05)
 - o Hiển thị toàn bộ danh sách các object Product.
 - o Thêm mới một object vào Firebase Realtime Database.
 - o Cập nhật thông tin của một object.
 - Xây dựng ứng dụng ReactJS có sử dụng AJAX để call các API trên để thực hiện xây dựng các giao diện để thực hiện các chức năng
 - o Hiển thị danh sách các object product, liệt kê đầy đủ các thông tin của Object, ở mỗi Object sẽ có một Button Edit
 - o Khi user click chọn vào Button Edit này, thì sẽ chuyển sang page Edit để cho phép:
 - User nhập thông tin mới của đối tượng hiện tại (tuy nhiên không được thay đổi objectID)
-

- Sau khi cập nhật thông tin xong, user click vào button Save thì ứng dụng sẽ gọi API để cập nhật lại thông tin mới của Object lên Firebase Database.
- Tương tự, nếu user click chọn Button Add New thì sẽ chuyển sang page Add New cho phép:
 - User nhập thông tin của một Object mới hoàn toàn (không nhập objectID).
 - Sau khi nhập xong, user click chọn button Save thì ứng dụng sẽ gọi API để lưu trữ lại thông tin của Object mới trên hệ thống Firebase Database.
 - Sau khi save xong thì ứng dụng sẽ quay lại màn hình danh sách và load lại danh sách với Object mới vừa được insert vào Firebase.
- *Lưu ý:*
 - **objectID** sẽ là mã được phát sinh tự động từ Firebase mỗi khi có object được lưu trữ vào.
 - Tái sử dụng lại bài tập 01 của Lab 05 để không cần phải xây dựng lại API.

Bài 2: (3 điểm)

- Tạo một project Firebase với tên Store Management.
- Tạo một Firebase Realtime Database có cấu trúc như sau:

```
{
  "categories": {
    "objectID": {
      "name": "mobile"
    }
  },
  "brands": {
    "objectID": {
      "name": "Apple"
    }
  }
}
```

- Tái sử dụng lại bài tập 02 của Lab 05 - Xây dựng API để thực hiện các yêu cầu sau:
 - Category:
 - Hiển thị toàn bộ danh sách các object Categories.
 - Thêm mới một object Category vào Firebase Realtime Database.
 - Cập nhật thông tin của một object Category.
 - Xóa một object Category với objectID được truyền vào.
 - Hiển thị thông tin của một object Category với objectID được truyền vào.
 - Brand:
 - Hiển thị toàn bộ danh sách các object Brands.
-

- Thêm mới một object Brand vào Firebase Realtime Database.
 - Cập nhật thông tin của một object Brand.
 - Xóa một object Brand với objectID được truyền vào.
 - Hiển thị thông tin của một object Brand với objectID được truyền vào.
 - Xây dựng ứng dụng React kết hợp AJAX để call API để thực hiện xây dựng giao diện cho ứng dụng trên với các chức năng được yêu cầu phát triển như sau:
 - Xây dựng các màn hình để thực hiện các chức năng quản lý Category như:
 - Liệt kê danh sách
 - Thêm mới
 - Cập nhật thông tin
 - Xóa
 - Xây dựng các màn hình để thực hiện các chức năng quản lý Brand như:
 - Liệt kê danh sách
 - Thêm mới
 - Cập nhật thông tin
 - Xóa
 - *Lưu ý:*
 - **objectID** sẽ là mã được phát sinh tự động từ Firebase mỗi khi có object được lưu trữ vào.
 - Tái sử dụng lại bài tập 02 của Lab 05 để không cần phải xây dựng lại API.
-

Bài 3: (4 điểm)

- Tạo một project Firebase với tên System Management.
- Tái sử dụng lại bài tập 03 của Lab 05. Tạo một Firebase Realtime Database có cấu trúc như sau:
 - o Thực hiện các công việc đối với user thuộc role_group là user:
 - Liệt kê thông tin tất cả các sản phẩm.
 - Liệt kê thông tin của các sản phẩm theo Brand với Brand Name được truyền vào.
 - Liệt kê thông tin của các sản phẩm theo Category với Category Name được truyền vào.
 - Hiển thị thông tin chi tiết của một Product với objectID được truyền vào.
 - o Thực hiện các công việc đối với user thuộc role_group là staff:
 - Thêm Brand mới.
 - Thêm Category mới.
 - Thêm một Product mới.
 - Cập nhật tên của một Brand.
 - Cập nhật tên của một Category.
 - Cập nhật thông tin của một Product.
 - Xóa một Brand.
 - Xóa một Category.
 - Xóa một Product.
 - o Thực hiện các công việc đối với user thuộc role_group là Admin:
 - Thay đổi role cho một user.
 - Cập nhật thông tin của user.
 - Thêm một user mới.
 - Xóa một user.
 - Liệt kê danh sách tất cả các user.
 - Hiển thị thông tin của một user với objectID truyền vào.
- Xây dựng ứng dụng ReactJS kết hợp với AJAX để thực hiện xây dựng giao diện cho ứng dụng này với các yêu cầu cụ thể như sau:
 - o Xây dựng màn hình Login cho phép user login vào hệ thống với tài khoản đã được lưu trữ sẵn trong Firebase Database (theo username và password) thông qua các API đã được tạo sẵn. Nếu tài khoản đăng nhập vào mà không thuộc các tài khoản đang được lưu trữ thì sẽ báo lỗi.
 - o Khi sau đăng nhập thành công với các loại tài khoản sau sẽ thực hiện các yêu cầu cụ thể sau:

```
{
  "categories": {
    "objectID": {
      "name": "mobile"
    }
  },
  "brands": {
    "objectID": {
      "name": "Apple"
    }
  },
  "products": {
    "objectID": {
      "name": "Galaxy S 21",
      "price": "1000",
      "brand": "brand_objectID",
      "category": "category_objectID"
    }
  },
  "role_group": {
    "users": {
      "objectID": {
        "username": "user_01",
        "password": "123456",
        "full_name": "Peter"
      }
    },
    "staff": {
      "users": {
        "objectID": {
          "username": "user_05",
          "password": "999888",
          "full_name": "Julie"
        }
      }
    },
    "admin": {
      "users": {
        "objectID": {
          "username": "admin_06",
          "password": "456789",
          "full_name": "Boss"
        }
      }
    }
  }
}
```

- Nếu user đăng nhập hiện tại thuộc role “users” thì:
 - Hiển thị danh sách sản phẩm - Product.
 - Mỗi sản phẩm sẽ có một button Detail. Khi user click vào button Detail này thì sẽ chuyển sang page Detail của một Product.
 - Xem thông tin chi tiết của một sản phẩm – Product, lưu ý sẽ phải load category_name và brand_name.
- Nếu user đăng nhập hiện tại thuộc role “staff” hoặc “admin” thì sẽ thực hiện các chức năng sau:
 - Hiển thị danh sách sản phẩm - Product.
 - Mỗi sản phẩm sẽ có một button Detail. Khi user click vào button Detail này thì sẽ chuyển sang page Detail của một Product.
 - Mỗi sản phẩm sẽ có một button Edit. Khi user click vào button Edit này thì sẽ chuyển sang page Edit cho phép Edit thông tin của sản phẩm – Product hiện tại.
 - Page Detail: Xem thông tin chi tiết của một sản phẩm – Product, lưu ý sẽ phải load category_name và brand_name.
 - Page Edit: cho phép cập nhật thông tin mới cho sản phẩm được chọn.
 - Ở màn hình danh sách Product, nếu user click chọn vào button Add New thì sẽ chuyển page Add New và cho phép người dùng điền thông tin của sản phẩm mới.

- *Lưu ý:*

- Các request được truyền về bắt buộc phải truyền theo username và password.
- Cần phải xây dựng một middleware để kiểm tra quyền của các request để xem có thuộc role để thực hiện các chức năng quy định theo role hay không?
 - Nếu thỏa thì next sang route xử lý chính.
 - Nếu không thỏa thì trả về lỗi 403 – Access Denied.
- **objectID** sẽ là mã được phát sinh tự động từ Firebase mỗi khi có object được lưu trữ vào.
- Tái sử dụng lại bài tập 03 của Lab 05 để không cần phải xây dựng lại API.
- Lưu ý ở các màn hình Edit và màn hình Add New thì các cần phải hiện ComboBox (Drop-down-list) của Category Object hay Brand Object để cho người dùng lựa chọn dễ dàng hơn.