**Module 5 – ReactJS**

**Ứng Dụng Quản Lý Khu nghỉ dưỡng Furama**

# Mục đích

Học viên sử dụng những kiến thức đã học ở module 5 để làm 1 website ứng dụng quản lý khách hàng và các dịch vụ của khu nghỉ dưỡng Furama.

# Công nghệ phải dùng

Sau khi kết thúc tất cả bài tập, học viên sẽ nắm được các kiến thức sau:

|  |
| --- |
| REACTJS |
| ES6 |
| JSX |
| React Component |
| State & Props |
| Hook (useState, useEffect) |
| React Form |
| Router |
| API Clients (axios, promise, async/await) |
| Redux |

# Mô tả hệ thống

Xây dựng hệ thống quản lý khu nghỉ dưỡng Furama tại thành phố Đà Nẵng. Hệ thống được mô tả như sau:

1. Khu nghỉ dưỡng Furama sẽ cung cấp các dịch vụ cho thuê bao gồm Villa, House, Room.

Tất cả các dịch vụ này sẽ bao có các thông tin: Tên dịch vụ, Diện tích sử dụng, Chi phí thuê, Số lượng người tối đa, Kiểu thuê (bao gồm thuê theo năm, tháng, ngày, giờ).

* Riêng Villa sẽ có thêm thông tin: Tiêu chuẩn phòng, Mô tả tiện nghi khác, Diện tích hồ bơi, Số tầng.
* Riêng House sẽ có thêm thông tin: Tiêu chuẩn phòng, Mô tả tiện nghi khác, Số tầng.
* Riêng Phòng sẽ có thêm thông tin: Dịch vụ miễn phí đi kèm.

Ngoài ra, resort còn cung cấp các dịch vụ đi kèm như massage, karaoke, thức ăn, nước uống, thuê xe di chuyển tham quan resort.

Dịch vụ đi kèm sẽ bao gồm các thông tin: Tên dịch vụ đi kèm, Đơn vị, Giá tiền

1. Hệ thống quản lý còn cho phép quản lý thêm thông tin của Employee resort. Thông tin Employee sẽ bao gồm: Họ tên Employee, Ngày sinh, Số CMND, Số ĐT, Email, Trình độ, Vị trí, Lương.

Trình độ sẽ lưu trữ các thông tin: Trung cấp, Cao đẳng, Đại học và sau đại học

Vị trí sẽ lưu trữ các thông tin: Lễ tân, phục vụ, chuyên viên, giám sát, quản lý, giá đốc.

Mỗi Employee sẽ thuộc một bộ phận làm việc: Hiện tại resort bao gồm các bộ phận: Sale – Marketing, Hành Chính, Phục vụ, Quản lý.

1. Hệ thống cho phép quản lý thêm thông tin của các Customer đến thuê và sử dụng các dịch vụ của Furama. Thông tin của Customer bao gồm: Họ tên, Ngày sinh, Giới tính, Số CMND, Số Điện Thoại, Email, Loại khách, Địa chỉ (resort sẽ phân loại Customer để áp dụng các chính sách phục vụ cho phù hợp).

Loại Customer bao gồm: (Diamond, Platinium, Gold, Silver, Member).

Customer có thể sử dụng các dịch vụ thuê Villa, House, Phòng và các dịch vụ đi kèm.

Đối với các dịch vụ thuê Villa và House thì Customer cần phải làm hợp đồng thuê với Furama.

Hợp đồng thuê sẽ bao gồm các thông tin: Số hợp đồng, Ngày bắt đầu, Ngày kết thúc, Số tiền cọc trước, Tổng số tiền thanh toán.

# Yêu cầu hệ thống

**Task 1:** Thiết kế prototype với các công nghệ sau:

**(thực hiện đến hết bài Tổng quan về React JS & JSX)**

- HTML, CSS, JS.

- Bootstrap.

1. Thiết kế giao diện có header, footer, navigation chung cho các trang.
2. Tạo prototype Danh sách các dịch vụ tại Furama (dạng card ở bootstrap, có phân trang, sử dụng dữ liệu cứng).

Lưu ý: Tương ứng với mỗi dịch vụ, sẽ có 2 button:

* Sửa: điều hướng đến trang Sửa thông tin dịch vụ.
* Xoá: khi click thì sẽ hiển thị modal xác nhận Xoá dịch vụ.



1. Tạo prototype Sửa thông tin dịch vụ (sử dụng dữ liệu cứng).
2. Tạo prototype Thêm mới dịch vụ.
3. Tạo prototype Danh sách các khách hàng tại Furama (dạng bảng, có phân trang). Tương tự như dịch vụ, thì ở mỗi record khách hàng sẽ có 2 button Sửa, Xoá.

Bao gồm cả modal xoá khách (sử dụng dữ liệu cứng).

1. Tạo prototype Sửa thông tin khách hàng (sử dụng dữ liệu cứng).
2. Tạo prototype Thêm mới khách hàng.
3. Tạo prototype Tạo hợp đồng Khách hàng thuê Dịch vụ ở Furama.
4. Tạo prototype Danh sách các hợp đồng tại Furama (dạng bảng, có phân trang). **Không** cần làm 2 button Sửa, Xoá.

**Task 2:** Tạo dự án ReactJS và các component - Tạo các component tương ứng các giao diện đã thiết kế ở Task 1:

**(thực hiện đến hết bài React Components, State & Lifecycle method)**

* Component Header.
* Body (prototype các trang, chọn 1 trang bất kì). Có thể tìm hiểu trước cách sử dụng thẻ <router-outlet>, để thay đổi uyển chuyển giữa các trang.
* Component Footer.

**Task 3:** Chuyển đổi từ dữ liệu cứng ở HTML sang dữ liệu cứng ở array.

**(thực hiện đến hết bài Hook)**

1. Tạo file facility.ts có kiểu dữ liệu interface hoặc class POJO dịch vụ.

Và mảng các đối tượng dịch vụ.

2. Dùng các directive hiển thị ra danh sách dịch vụ trên component danh sách dịch vụ.

3. Tạo file customer.ts có kiểu dữ liệu interface hoặc class POJO khách hàng.

Và mảng các đối tượng khách hàng.

4. Dùng các directive hiển thị ra danh sách khách hàng trên component danh sách khách hàng.

5. Tạo file contract.ts có kiểu dữ liệu interface hoặc class POJO hợp đồng

Bao gồm các thông tin sau:

* Mã hợp đồng
* Thông tin khách hàng làm hợp đồng (object Customer)
* Thông tin dịch vụ mà khách hàng thuê (object Facility)
* Ngày làm hợp đồng.
* Ngày kết thúc hợp đồng.
* Tiền đặt cọc.

Và mảng các đối tượng hợp đồng.

6. Dùng các directive hiển thị ra danh sách hợp đồng trên component danh sách hợp đồng.

**Task 4:** Thực hiện validate dữ liệu.

**(thực hiện đến hết bài React Form & Router)**

1. Viết chức năng thêm mới cho form thêm mới và chỉnh sửa khách hàng đảm bảo các yêu cầu sau:

* Tên khách hàng không được chứa số. Và các kí tự đầu tiên của mỗi từ phải viết hoa.
* Số điện thoại phải đúng định dạng 090xxxxxxx hoặc 091xxxxxxx hoặc (84)+90xxxxxxx hoặc (84)+91xxxxxxx.
* Số CMND phải đúng định dạng XXXXXXXXX hoặc XXXXXXXXXXXX (X là số 0-9).
* Địa chỉ email phải đúng format.
* Ngày sinh, sử dụng date picker để người dùng có thể chọn ngày thuận tiện. Phải validate xem ngày giờ có hợp lệ hay không (đúng định dạng ngày giờ theo format DD/MM/YYYY).

1. Viết chức năng thêm mới cho form thêm mới và chỉnh sửa dịch vụ đảm bảo các yêu cầu:

* Tất cả các trường đều bắt buộc nhập.
* Tên dịch vụ không được chứa số.
* Diện tích hồ bơi, Số tầng phải là số nguyên dương.

1. Viết chức năng thêm mới cho form thêm mới hợp đồng cần đảm bảo các yêu cầu:

* Ngày làm hợp đồng, Ngày kết thúc sử dụng sử dụng date picker để người dùng có thể chọn ngày thuận tiện. Phải validate xem ngày giờ có hợp lệ hay không (đúng định dạng ngày giờ theo format DD/MM/YYYY).
* Tiền đặt cọc phải là số nguyên dương.

1. Khi click vào nút xóa (hoặc biểu tượng xóa) trên các dòng ở các màn hình danh sách thì phải hiển thị modal xác nhận việc người dùng có chắc chắn muốn xóa dữ liệu hay không.

Nếu người dùng chọn Yes thì xóa dữ liệu ở dòng đó. (Ở đây xóa dữ liệu ở array).

**Task 5:** Phân tách chức năng các đối tượng thành từng module (folder) riêng lẻ.

**(thực hiện đến hết bài Tổng quan về Next.js)**

Mỗi module (folder) sẽ bao gồm các thành phần:

* Các component CRUD.
* Các interface, class POJO của mỗi object.
* Các service (các array để chứa dữ liệu của mỗi object).
* Các router để thay đổi nội dung giữa các component CRUD.

Kết quả sẽ được 3 module (folder) như bên dưới:

* Module Facility (dịch vụ).
* Module Customer (khách hàng).
* Module Contract (hợp đồng)

**Task 6:** Sử dụng dữ liệu từ API.

**(thực hiện đến hết bài API Clients)**

Ở service của mỗi module, chuyển đổi toàn bộ array thành các dữ liệu ở JSON.

Bằng cách sử dụng axios để kết nối với Mock API.

Lưu ý: Các bạn tự cài đặt Mock API và file JSON data để truy xuất, lưu trữ dữ liệu.

**Task 7\*:** Sử dụng dữ liệu từ DB thông qua việc kết nối đến Webservice (được viết bởi Spring).

Thực hiện chuyển đổi các chức năng sau, từ kết nối đến Mock API thành kết nối đến Webservice (ở Spring):

* Danh sách khách hàng.
* Thêm khách hàng.
* Sửa khách hàng.
* Xoá khách hàng.

Lưu ý: Với việc hoàn thành phần task này thì sẽ được +5 điểm vào bài thi thực hành cuối Module, bằng cách báo lại và thuyết trình với instructor/tutor. Nội dung thuyết trình:

* Flow (luồng di chuyển) giữa ReactJS và Spring.
* Instructor/Tutor sẽ vấn đáp một vài câu hỏi để xác nhận lại việc hiểu của bạn.