**Chapter 1: Giới thiệu ReactJS**

* **Multi page app:**

- Là 1 phương thức làm web cũ. Được áp dụng rộng rãi từ xưa đến nay.

- Đối với hầu hết mọi hành động thay đổi những gì trình duyệt hiển thị, ứng dụng web sẽ điều hướng bạn đến một trang hoàn toàn khác. Ví dụ: thêm, sửa, xóa, chuyển trang, tìm kiếm, …

- Đây là một vấn đề lớn làm giảm performent của trang web, ngoài những trải nghiệm người dùng không tốt người dùng sẽ thấy khi các trang bị reload lại khi mỗi lần thao tác trên web. Ví dụ: <https://tiki.vn/>

* **Single page app:**
* Là phương thức làm web mới được áp dụng trong khoản thời gian gần đây và đang được phát triển mạnh mẽ. Những trang web hiện đại đều tuân thủ mô hình Single page app.
* Mô hình này cho phép bạn hoạt động như 1 ứng dụng mà không cần phải điều hướng đến các trang khác nhau hay reload lại trang để thực hiện hành động nào đó. Nó giống như 1 ứng dụng điện thoại vậy đó. Ví dụ: <https://phongvu.vn/>

- Những khó khăn gặp phải khi phát triển Single page app:

1. Bạn sẽ phải dành phần để giữ cho dữ liệu của bạn đồng bộ được với UI. Ví dụ như thêm 1 item vào list thì cũng phải thêm phần tử HTML tương ứng.

2. Thao tác với DOM thật sự rẩt chậm. Ví dụ: Khó khăn trong việc tìm kiếm item để xóa trong list.

3. Cực kì khó áp dụng các thư viện hỗ trợ giao diện.

* **Làm quen với React:**
* Được Facebook tạo ra để giải quyết các vấn đề khó khăn khi làm việc với single page app trong quá trình phát triển nền tản của mình và dần dần trở thành thư viện tiên phong trong việc xây dựng ứng dụng single page app.
* React không chỉ giải quyết những thiếu sót trong quá trình phát triển single page app mà còn thay đổi cách chúng ta nghĩ về việc xây dựng các ứng dụng web.
* **Các tiện ích của React:**
* *Quản lý trạng thái UI tự động:*

+ Đối với React bạn không phải quan tâm vấn đề người dùng bắt đầu ở trạng thái nào hay người dùng thực hiện các hành động nào để thay đổi giao diện của trang web.

* *Thao tác với DOM 1 cách nhanh chóng:*

+ Vì như đã nói, việc sửa đổi trực tiếp DOM là rất chậm nên React sẽ không làm thế. Thay vào đó nó đẩy vào bộ nhớ 1 DOM ảo. Khi vào trang web lần đầu tiên, trình duyện sẽ load hết toàn bộ source của bạn rồi khi nào bạn cần sẽ lôi nó ra.

* *Tạo ra sự tổng hợp các thành phần UI lại với nhau:*

+ Thay vì coi các element trong ứng dụng của bạn là một khối nguyên khối, React khuyến khích bạn chia các element của mình thành các thành phần nhỏ hơn và nhỏ hơn.  
+ Khi đó chúng ta có lợi thế là tái sử dụng và thực hiện thay đổi trên từ thành phần nhỏ đó dễ dàng hơn. Ví dụ: Header, Footer có thể tái sử dụng ở nhiều trang khác nhau trong cùng 1 trang web.

* *Được xây dựng hoàn toàn dự trên JavaScript:*

+ Bằng cách xây dựng UI của bạn hoàn toàn bằng JavaScript, bạn có thể sử dụng tất cả các chức năng phong phú mà JavaScript cung cấp để thực hiện tất cả các loại templates của bạn.

+ Bạn chỉ bị giới hạn bởi những gì JavaScript hỗ trợ, không bị giới hạn bởi khung khuôn mẫu của bạn.

* *Là toàn bộ phần V (View) trong mô hình MVC:*

+ Nhắc lại MVC:

*Model (M):*

*Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller.*

*View (V):*

*Đây là phần giao diện dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website.*

*Controller (C):*

*Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.*

+ React hoạt động chủ yếu trong lớp View, xoay quanh việc giữ cho các elements của bạn được cập nhật.