# BÀI 65 : React là gì? Tại sao lại chọn React? So sánh React vs Angular vs Vue

## 1 . React là gì? Tại sao lại chọn React?

ReactJS là một framework JavaScript dùng để xây dựng các **Single Page Application (SPA)**.

SPA là những trang web mà không bị reload khi chúng ta chuyển trang. Ví dụ: Facebook, Twitter, Instagram, YouTube, Gmail,...

Ngoài React thì còn có rất nhiều JS Framework có thể tạo được SPA như Angular, Vue, Svelte,...

### Tại sao lại chọn React?

Chúng ta sẽ so sánh React với 2 framework lớn hiện tại là Angular và Vue.

*Độ khó:*

* React và Vue theo mình sẽ dễ tiếp cận hơn cho những bạn mới bắt đầu học về các framework SPA.
* Angular là framework khá khó nếu mới bắt đầu vì bạn phải học TypeScript cùng những concept khá phức tạp như RXJS, Dependency Injection,...

*Sự phổ biến:*

* React là framework SPA phổ biến nhất hiện nay, số lượt tải hàng tháng gấp 3 lần Vue và Angular cộng lại.
* Hệ sinh thái xung quanh React cũng rất lớn, được rất nhiều các công ty trên thế giới tin dùng.
* Nhu cầu việc làm của React cũng lớn hơn rất nhiều so với Vue và Angular.

*Tốc độ:*

* Về tốc độ của React, Angular, Vue nhìn chung thì cũng tương đương nhau, chúng ta không cần quá quan tâm về điều này.
* Kích thước file Build của React, Angular, Vue cũng phụ thuộc khá nhiều vào những thứ chúng ta implement vào như thư viện sử dụng nên cũng khó mà đánh giá.

*Hệ sinh thái:*

* Như mình đã nói thì hệ sinh thái (HST) React rất vượt trội, nếu bạn học React bạn cũng có thể viết được mobile với React Native hay desktop app với Electron.
* Next.js là một framework React hỗ trợ server-side rendering mạnh nhất hiện nay.

## 2. So sánh SPA vs MPA

* MPA là Multiple Page Application, tức là những website truyền thống. Khi chuyển trang thì sẽ load lại toàn bộ trang web.

*Độ khó học:*

* SPA khó hơn so với MPA khi bạn phải học thêm một đống thứ xung quanh JavaScript framework.

*SEO:*

* MPA SEO tốt hơn so với SPA vì trả về source HTML ngay khi load trang web, còn SPA thì phải mất thời gian mới render ra HTML.
* Những bot crawler hiện nay không đọc tốt những trang web mà SPA.
  + Nhưng vẫn có cách cải thiện điểm yếu này là render HTML ngay tại server luôn rồi trả về tương tự MPA. Chúng ta có thể dùng Next.js thay vì React thuần.

## UX:

* SPA tăng trải nghiệm người dùng vì không phải tải lại toàn bộ trang web.
* MPA tải lại cả trang web mỗi khi chuyển trang đem lại trải nghiệm không thân thiện.

## Thân thiện dev

* SPA giúp phân chia rõ ràng code giữa frontend và backend => phát triển dễ dàng.
* SPA có thể tái sử dụng các component dễ dàng.
* MPA thì không có sự phân chia rõ ràng giữa frontend và backend, backend đôi khi cũng phải xử lý những công việc của frontend.

## Tốc độ tải trang

* Load lần đầu: MPA nhanh hơn so với SPA.
* Những lần chuyển trang tiếp theo: SPA nhanh hơn MPA.
* Server bên SPA sẽ được giảm tải hơn khi so với MPA.
* Server thiết kế cho SPA cũng có thể dùng được cho mobile => tiết kiệm thời gian.

## 3. Những cách setup một dự án React

1. Chèn file JavaScript tương tự như cách chúng ta dùng jQuery.  
   Cách này dùng khi chúng ta đã có một website từ trước và muốn implement React vào một phần nhỏ trong website, cách này sẽ có nhiều hạn chế.
2. Dùng Create React App.  
   Đây là tool tạo nhanh một project React do Facebook phát triển => Cách này đáp ứng được hầu hết nhu cầu của chúng ta khi phát triển dự án.
3. Tự build project React dựa trên Webpack hoặc Vite.  
   Cách này giúp chúng ta tùy chỉnh sâu hơn theo nhu cầu.

# Bài 68: Những thứ bạn cần cài đặt để có một môi trường hoàn hảo code React

Cài đặt môi trường code React

## 1. Môi trường hoàn hảo để code

1. Node.js để run server React dưới local.
2. VS Code để code:
   * Cài thêm extension Icon Theme cho VS Code (mình dùng Material Icon Theme).
   * Prettier để format code.
   * ESLint để quản lý tiêu chuẩn code.
3. Trình duyệt Chrome với các extension:
   * React Developer Tool.
   * Redux Dev Tool extension.

A screenshot of a computer

Description automatically generated

1. Git: Tạo repo trên GitHub để quản lý source code.

## 2. Khởi tạo Project với CRA (Create React App)

Ép xuống phiên bản 18 : npx create-react-app my-app --template cra-template@1.0.1

## 3. Giới thiệu về cấu trúc folder của React.js

## 4. Sử dụng Prettier và ESLint trong React.js

### 4.1. Setting Prettier on User

Editor: Default Formatter ⇒ Chọn Prettier

Editor: Format On Save ⇒ true

Files: EoL ⇒ auto

Nếu như các bạn muốn setting mỗi workspace hiện tại thôi (nghĩa là cái folder hiện tại) thì các bạn setting bên workspace.

### 4.2. Tạo file .editorconfig để chia sẻ một số setting giữa các editor với nhau

[\*]

indent\_style = space

indent\_size = 2

### 4.3. Tạo file .prettierrc để chia sẻ setting Prettier giữa các editor

{

"arrowParens": "always",

"semi": false,

"trailingComma": "none",

"tabWidth": 2,

"endOfLine": "auto",

"useTabs": false,

"singleQuote": true,

"printWidth": 120,

"jsxSingleQuote": true

}

### 4.4. Cài các devDependencies hỗ trợ Prettier và ESLint trên terminal

Ở trên thì chúng ta mới chỉ setting Prettier cho editor (tức là UI), bây giờ là cho terminal. Editor thì có thể mỗi máy sẽ kiểm tra khác nhau nhưng trên terminal thì không, setting trên terminal sẽ giúp code thống nhất.

Bash :

npm i prettier eslint-plugin-prettier eslint-config-prettier -D

Trên là những plugin cần thiết để setting Prettier và ESLint cho CRA. Các bạn nếu dùng yarn thì chạy yarn add :

Bash:

yarn add prettier eslint-plugin-prettier eslint-config-prettier -D

### 4.5. Tạo file .eslintrc để setting ESLint

{

"extends": ["react-app", "prettier"],

"plugins": ["react", "prettier"],

"rules": {

"prettier/prettier": [

"warn",

{

"arrowParens": "always",

"semi": false,

"trailingComma": "none",

"tabWidth": 2,

"endOfLine": "auto",

"useTabs": false,

"singleQuote": true,

"printWidth": 120,

"jsxSingleQuote": true

}

]

}

}

### 4.6. script bạn cần thêm vào file package.json

{

"lint": "eslint --ext js,jsx,ts,tsx src/",

"lint:fix": "eslint --fix --ext js,jsx,ts,tsx src/",

"prettier": "prettier --check \"src/\*\*/\*(.jsx|\*.js|\*.tsx|\*.ts|\*.css|\*.scss)\"",

"prettier:fix": "prettier --write \"src/\*\*/\*(.jsx|\*.js|\*.tsx|\*.ts|\*.css|\*.scss)\""

}

Lúc này bạn chỉ cần chạy:

* npm run lint: Kiểm tra lỗi ESLint
* npm run lint:fix: Sửa lỗi liên quan đến ESLint (đôi lúc có những lỗi bạn phải tự fix bằng tay)
* npm run prettier: Kiểm tra lỗi Prettier format
* npm run prettier:fix: Tự động sửa lỗi Prettier format

### 4.7. Thêm file .prettierignore và .eslintignore để ignore những file bạn không muốn Prettier và ESLint format

Cú pháp viết trong những file này tương tự như trong file .gitignore. Các bạn có thể tìm lại trong file GitCanBan.md trong thư mục GitCanBanDenNangCao nhé.

## 5. CICD và deploy với Vercel (Bài 72: Setting CICD tự động deploy với Vercel

## ):