**1.Why React ?(** **React Có Gì Hot So Với Vue và Angular)**

**-** Vì react có nhiều lượt tải trên npm nhiều hơn vue với angular.

- Nó không load lại website.

**2. Class** vs Hook (function component).  **Redux** vs Context API ?**( Lựa Chọn Công Nghệ Sử Dụng Với React)**

Hook sẽ giúp tách từng phần xử lí logic của một component ném ra một file riêng biệt giúp có thể tái sử dụng code. Redux là sử dụng cả hai cái là Class và Hook (cái nào tiện thì sử dụng).

**3. Setup env(Cài Đặt Môi Trường và Công Cụ Cho Dự Án React.JS)**

Cài đặtVisual Studio Code, NodeJS, Git

**3.1 Hello world with React(Cài đặt Project Local)**

<https://github.com/KietHo1/ReactJS-Basic-KietHo1-Part1>

Edit src/App.js

**3.2 Source code on Github(SOS Thì Làm Sao)**

Cài đặt code từ github

**3.3 Structure React Project(Cấu Trúc Project React.JS - Files Structure)**

**-** Webpack: để nén chương trình từ 1MB thành vài KB và làm tăng hiệu năng(performance).

- Babel: để biên dịch nhiều phiên bản Javascript khác nhau.

**-** Cấu trúc project react gồm: assets(ảnh và audio chia sẻ public), components(tái sử dụng code), routes, services(chứa các API), stores, styles(chứa các file scss), utils, views

- Import scss module: npm i –save sass

- Video hướng dẫn phẩn tổ chức file(28:16 đến 33:16): https://www.youtube.com/watch?v=JsQfIjzaBNE&list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

**4. React components(Viên Gạch Tạo Nên Bố Cục Của React.JS)**

- JSX là gì? Vd: const element = <h1> Hello World </h1>;

- Component là những khối gạch xây dựng nên một trang web. (return ra nhiều html là một component)

- Component có 2 loại: **Class** và Function( funtion và arrow)

- Để sử dụng Class component: export {}(nhiều cái) hoặc export default(một cái)

- Khai báo Full: <MyComponent> </MyComponent> (render con của nó)

- Khai báo **Shortcut**: <MyComponent /> ( component không có con)

**5. State(Kiểm Soát Trạng Thái Ứng Dụng React.JS)**

- Có thể chèn Javascript vào trong html bằng dấu {}

- Chỉ return ra được một cái <div> </div>

- Thẻ div có className chứ không phải là class

- Nhiều thẻ div có thể chứa trong thẻ <React.Fragment> </React.Fragment> hoặc thẻ <> </>

- Khai báo biến trong JSX: let name = ‘Eric’ và sử dụng biến vừa khai báo: My name is {name}

- State là những cái gì mà nó thay đổi. Khi thay đổi giá trị của biến thì state thay đổi theo(đó chính là điều kì diệu của state)

**6. React dev tool(Công Cụ Hữu Ích Của Lập Trình Viên)**

- Search google: React developer tool

- Chỉ quan tâm đến tab Component (chuột phải 🡪 Inspect)

**7. DOM Events(Xử Lý Tất Tần Tật Events Với React.JS)**

- DOM là bố cục của trang html (các phẩn tử cấu tạo nên một trang web hay là các thành phần trong tab element)

- DOM Events là các sự kiện trên website

- Search google “DOM Events” để xem thêm các sự kiện

**8. Changing State(Thay Đổi State và Re-render của React.JS)**

- Thay đổi cái gì thì code vào this.setState 🡪 Thư viện sẽ tự động ghép vào(merge) cho đầy đủ trang web.

**9. Intro to Forms(Thu Thập Dữ Liệu Inputs HTML)**

-flex-direction: column (các div xuống hàng).

- ngôn ngữ server-side(backend): từ giao diện html gửi tới server khác với react: handle nó rồi gọi đến api(không phụ thuộc về phía backend).

**10. Nesting components(QUAN HỆ CHA CON của REACT)**

- Nesting nghĩa là lồng nhau

- Chỉ truyền được prop từ component cha sang component con và không có ngược lại.

**11. Props(Truyền Data từ Cha Sang Con)**

- Chỉ truyền được prop từ component cha sang component con và không có ngược lại. Nhưng có thể dùng function để gọi component cha từ component con.

**12. Outputting Lists(Renders Nhiều Data với Props)**

Renders Nhiều Data với Props

**13. Stateless/StateFull Components (Function/Class Components)**

- Video này làm về function component( khác với class component ở những video trước)

- Stateless là không có state (dùng prop)  
 - Statefull là có state

- Không dùng biến this vì không có class(truyền vào prop và dùng prop)

**14. Conditional Output(Sử dụng Điều Kiện Với React.JS)**

- Sử dụng button Show và Hide

**15. Split Components(Chia cắt Components Trong React.JS)**

- Chia cắt component trong ReactJS

**16. Functions as props(Truyền Functions Từ Cha sang Con**

- truyền data từ component cha sang component con

- cách truyền function từ parent Component vào child Component như props (fuction as props), chỉ ghi tên hàm và ko thêm "( )"

**17. Deleting data(Xóa dữ liệu)**

- Frontend không thể lưu trữ dữ liệu

- Backend sẽ lưu trữ dữ liệu và khi reload trang thì dữ liệu sẽ được đưa từ database(backend) trở lại trang web

**18. Recap & Virtual DOM(Tổng kết Chặng Đường & Virtual DOM)**

**-** https://www.youtube.com/watch?v=R79CtLxih-A&list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy&index=24

**19. CSS Files(CSS với React Sao Cho Nó Đúng)**

- Đi làm dùng scss(sass) chứ không dùng css

**20. Lifecycle Methods(Vòng Đời Ứng Dụng)**

- https://projects.wojtekmaj.pl/react-lifecycle-methods-diagram

- https://legacy.reactjs.org/docs/react-component.html

**21. To Do Apps (CRUD - create/read/update/delete)(** **Xây Dựng ToDo APP PRO với React)**

**21.1 Create/Read data todos**

**21.2 Update/Delete todo**

**22. The react routers(Điều Hướng Trang Với React)**

- Chuyển trang bị load lại

**23. Links & NavLinks(Links & NavLinks Trong React.JS Routers)**

- Chuyển trang không bị load lại

**24. Redirect(chuyển trang)**

https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

**25. Higher Order Components(HOC)**

https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

**26. Using axios(Sử Dụng Axios Để Gửi Request API từ React.JS)**

https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

**27. Route parameters(Sử Dụng Tham Số Trên URL)**

https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

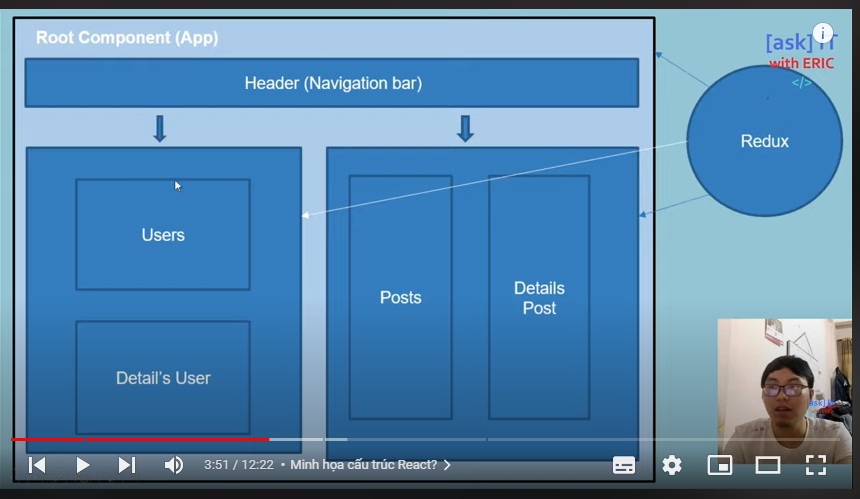
**28. Switch tags**

https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

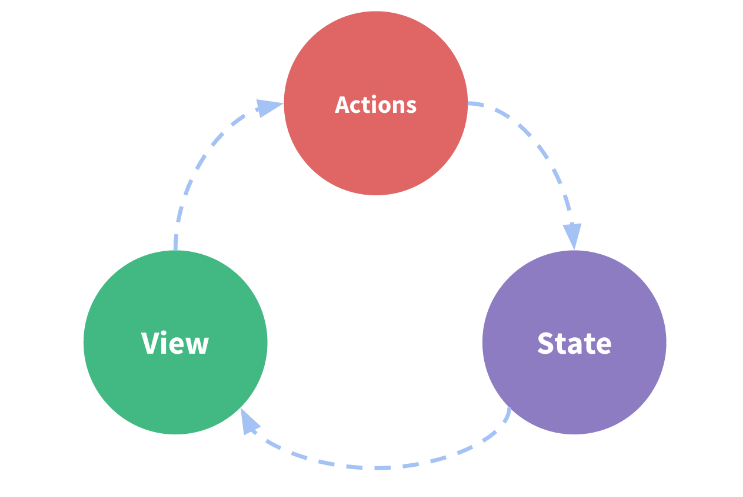
**29. Importings images(Sử Dụng Hình Ảnh Trong Ứng Dụng React)**

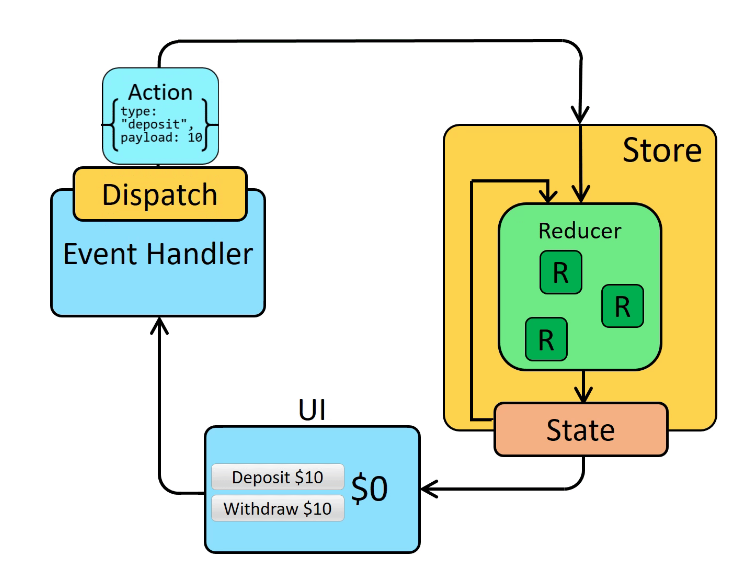
https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

**30. Introduce to Redux(Why Redux ? Redux Có Gì Hay So Với Context API & Local Storage)**



**31. Redux Stores, Actions, Reducers(Học Redux)**





<https://redux.js.org/tutorials/essentials/part-1-overview-concepts>

**32. Setting up Redux in React app(Setup Redux Vào Dự Án React)**

https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

**33. Mapping State (Redux) to Props (React)**

https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

**34. Map Dispatch to Props (React) - Redux Dev Tool(Áp dụng redux để thêm/xóa dữ liệu react)**

https://www.youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy

**35. What's Next ?(Tổng kết khóa học)**

**Chú ý: Video 32, 33, 34 cài đặt** [**react-redux@7.2.5**](mailto:react-redux@7.2.5) **và redux@4.1.1**