**Notes**

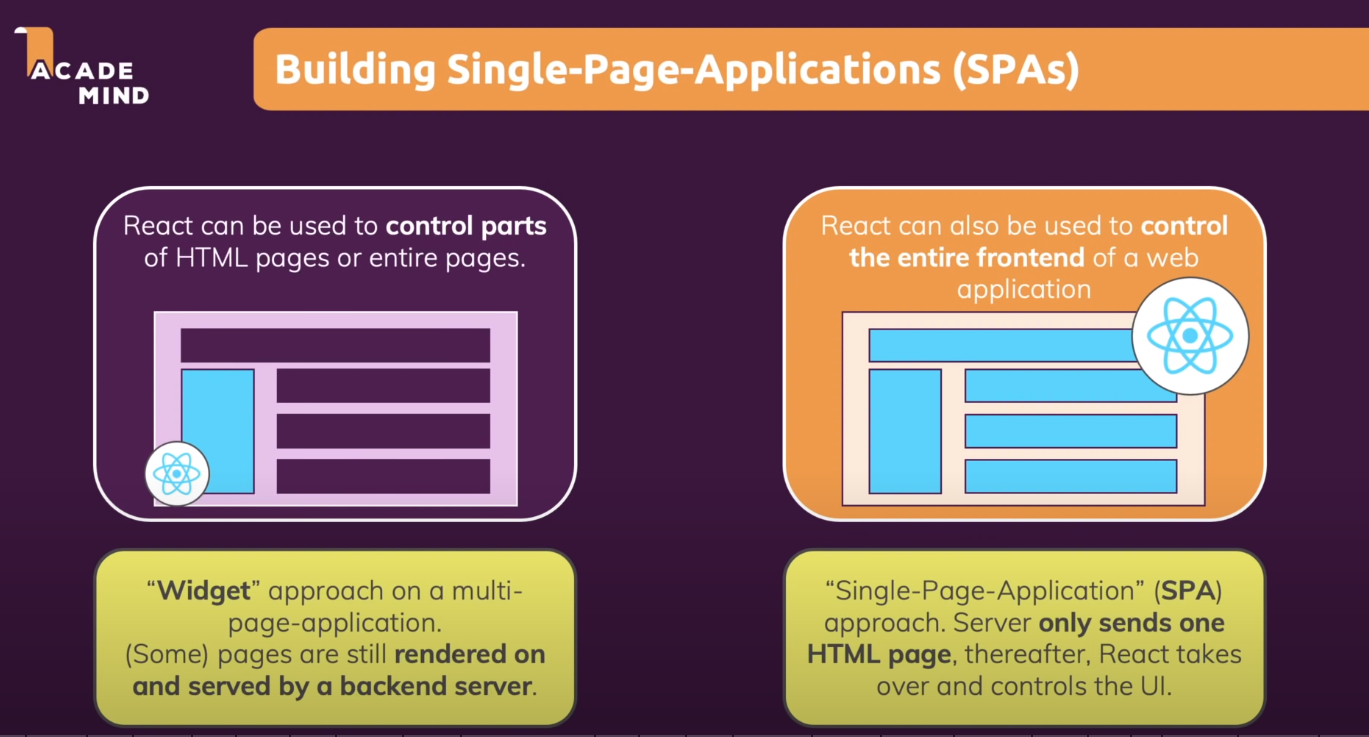
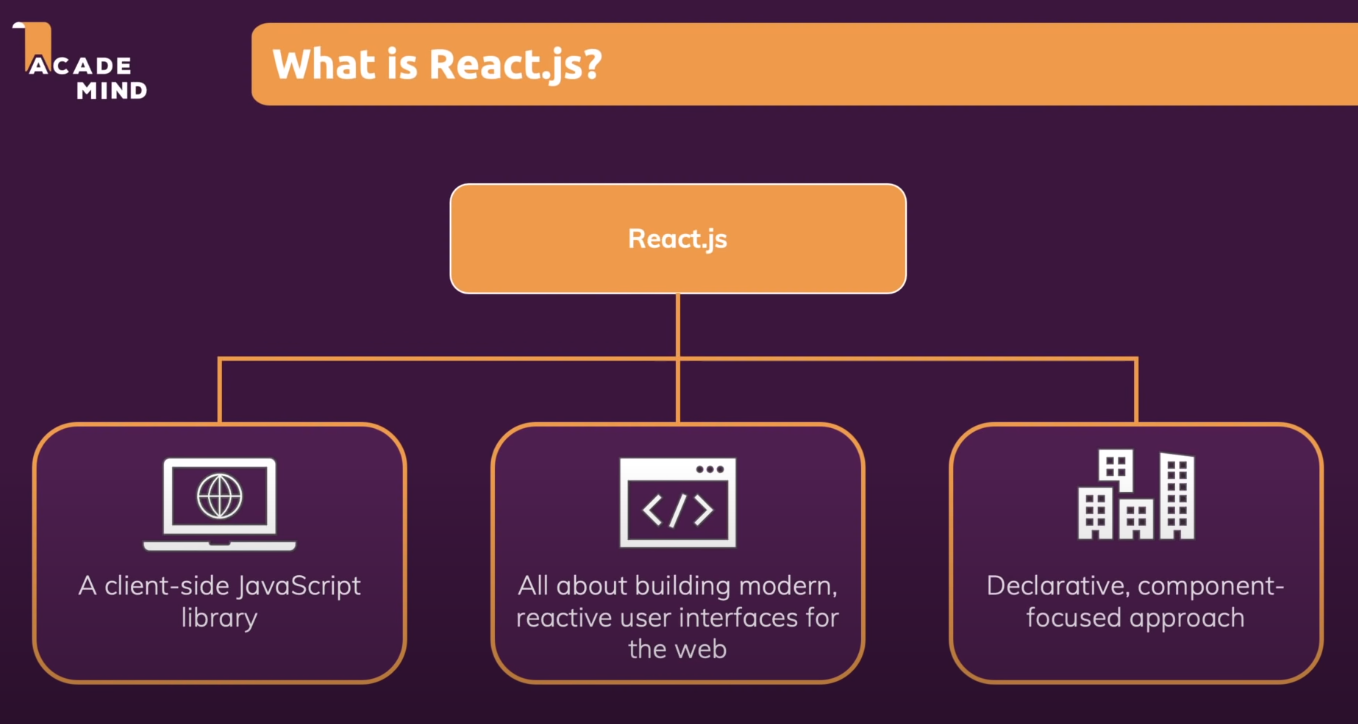
What is React:

* It is JS Library for building UI
* First react was class components and now react (*version 16.8*) updated to functional components
* **Next.js** is a frame work of React

Why we use React:

* Don’t need to refresh page or never request a new html page request (which make smother UI and better User)
* Because it is JS library (and JS work on user’s browsers so in **REACT** with this behavior we manipulate web pages after they were loaded {that allow us to build highly interactive modern UI for website} )
* And React do this manipulation with the help of **Virtual** **DOM**
* With the use of **React (**and other JS frameworklike **Angular** and **VUE )** we built Single Page Application (**SAP**)
* React help us to make single page application

React Features & Working on (SPA & Widget):



React vs Angular vs VUE:

|  |  |  |
| --- | --- | --- |
| **React** | **Angular** | **VUE** |
| **According to Bard** | | |
| Pros:   * Easy to learn * Flexible and customizable * Large community and ecosystem * Fast performance   Cons:   * Can be difficult to debug complex applications * Requires a good understanding of JavaScript | Pros:   * Comprehensive and full-featured * Opinionated and easy to follow architecture * Good performance and scalability * Large community and ecosystem   Cons:   * Can be difficult to learn * Can be overkill for small or simple applications * Requires a good understanding of TypeScript | Pros:   * Easy to learn * Lightweight and flexible * Good performance and scalability * Large community and ecosystem   Cons:   * Not as mature as React or Angular * Not as comprehensive as Angular |
| **According to ChatGpt** | | |
| Pros:   * Component-based architecture that encourages reusability. * Extensive community support and a vast ecosystem of libraries and tools. * React Native allows for building mobile apps using the same knowledge. * Virtual DOM for efficient rendering.   Cons:   * React is primarily a view library, so you might need to integrate other libraries for routing, state management, and other functionalities. | Pros:   * Strongly opinionated, which can lead to consistency and maintainability. * Extensive features, including dependency injection, a powerful template system, and an integrated development environment. * TypeScript is the default language, which enforces type safety.   Cons:   * A steeper learning curve, especially for newcomers to web development. * A larger bundle size compared to React or Vue. | Pros:   * Easy to learn and integrate into existing projects. * Vue offers a simple and flexible API, making it approachable for developers of varying skill levels. * Single-file components make it easy to organize code.   Cons:   * Smaller community and ecosystem compared to React and Angular. * Vue's adoption may not be as high in large enterprises as Angular. |

Props:

* It is important for building reusable components.

State:

* It is important for changing what we see on the screen dynamically.
* For adding and removing elements for changing text
* Also for showing, hiding and overlay the components.