**ReactJS**

**Popularity**

* ReactJS is an open source JavaScript library designed by Facebook
* Today, it is a very popularity front-end framework, because of its simplicity and flexibility
* It is estimated that more than 1,300 developers and 94,000 websites utilize ReactJS

**Strengths**

* Easy to learn
* Developers only need to familiarize themselves with HTML and CSS
* Simple framework
* Well-defined life cycle
* Component-based approaches
* JSX allows easy combination of JavaScript and HTML
* Reusable code components
* Able to break down complex components into smaller reusable components
* Able to complete projects more quickly
* One-direction data flow in ReactJS provides stable code
* Child elements will not affect the parent data
* Props make it possible to populate components using custom data
* Able to pass custom data to a specific UI
* Allows developers to fill blanks when rendering JSX components
* State stores all changes in one section
* All details are sourced from a single location
* Web applications become very simple
* Virtual DOM makes the user's experience better and developer’s work faster
* Detects when there has been a change in data
* Knows exactly when to re-render the web application
* Fast run-time performance
* Fast to display a large amount of components
* Open-source and constantly developing
* Easy debugging and testing
* SEO-friendly

**Weaknesses**

* Poor documentation
* Small contributing community
* Becomes relatively slow when there are very many users
* Requires manual processing of data changes

**What developers mostly use it for?**

* Develop web applications with high quality user interfaces
* Develop scalable, fast, and simple web applications
* Develop large web-based applications that change data while reloading pages
* Online businesses
* Search Engine Optimization (SEO)

**What kind of websites?**

* Social media
* Facebook
* Instagram
* Reddit
* WhatsApp
* Online businesses
* Airbnb
* Netflix
* PayPal
* Postmates
* Uber

**Ruby on Rails**

**Popularity**

* Ruby on Rails or Rails is a back-end web application framework that uses Ruby under the MIT license
* Rails peaked in popularity in 2011 where its main selling point was the velocity of its workflow in creating web applications.
* Innovative features include seamless database table creations, migrations and scaffolding of views to enable rapid application development.
* Ruby on Rails influence on other web frameworks are very apparent today from Django, Catalyst, and Laravel all borrowing ideas from Rails.
* According to a study conducted in June 2017 Rails had the highest mention in job postings from the San Francisco Bay Area with a count of 1066 . Django comes in second with 636

**Strengths**

* Libraries
* Libraries or Gems as they are called in Rails are extremely varied. You can find a game on just about anything from writing automated tests to coding style
* Tooling
* W Rails provides fantastic tools that can help speed up web application delivery. One tool helps provide a standard structure for web apps where all the common patterns are taken care of for you
* Component-based approaches
* JSX allows easy combination of JavaScript and HTML
* Ruby on Rails Community
* The Rails community is filled with talented engineers and developers
* The community in turn is incredibly interested in the advancement of the framework and provides plenty of knowledge and resources .
* This means that what you are creating with Rails is most likely already created and so development will be more efficient due to the number of resources the developers can reference
* Promotes Industry Standards
* The central pillar of Rails is DRY(Don’t Repeat Yourself) principle that ensures a clear separation of concerns and maintainability of your application.
* Framework embraces the principle of ‘convention over configuration’, which Rails defaults to a set of conventions that specify the best way of doing things.
* Rails is also built around MVC(Model-View-Controller) philosophy that promotes modularity and extensibility of your applications.
* Fast to display a large amount of components
* Open-source and constantly developing
* Easy debugging and testing

**Weaknesses**

* Slow runtime speed and performance
* Lack of flexibility
* High cost of wrong decisions

**What developers mostly use it for?**

* Develop web applications with high quality user interfaces
* Develop scalable, fast, and simple web applications
* Develop large web-based applications that change data while reloading pages
* Online businesses

**What kind of websites?**

* Airbnb
* Etsy
* Shopify
* Github
* Facebook